Как подключить к Apache Spark проприетарный источник данных: теория, практика, грабли, костыли

Александра Белоусова
Яндекс.Go

О чём я буду говорить

- Мотивация и постановка задачи
- Быстрое и простое решение
- Сложное производительное решение
- Грабли и костыли

Что надо было сделать?

YT

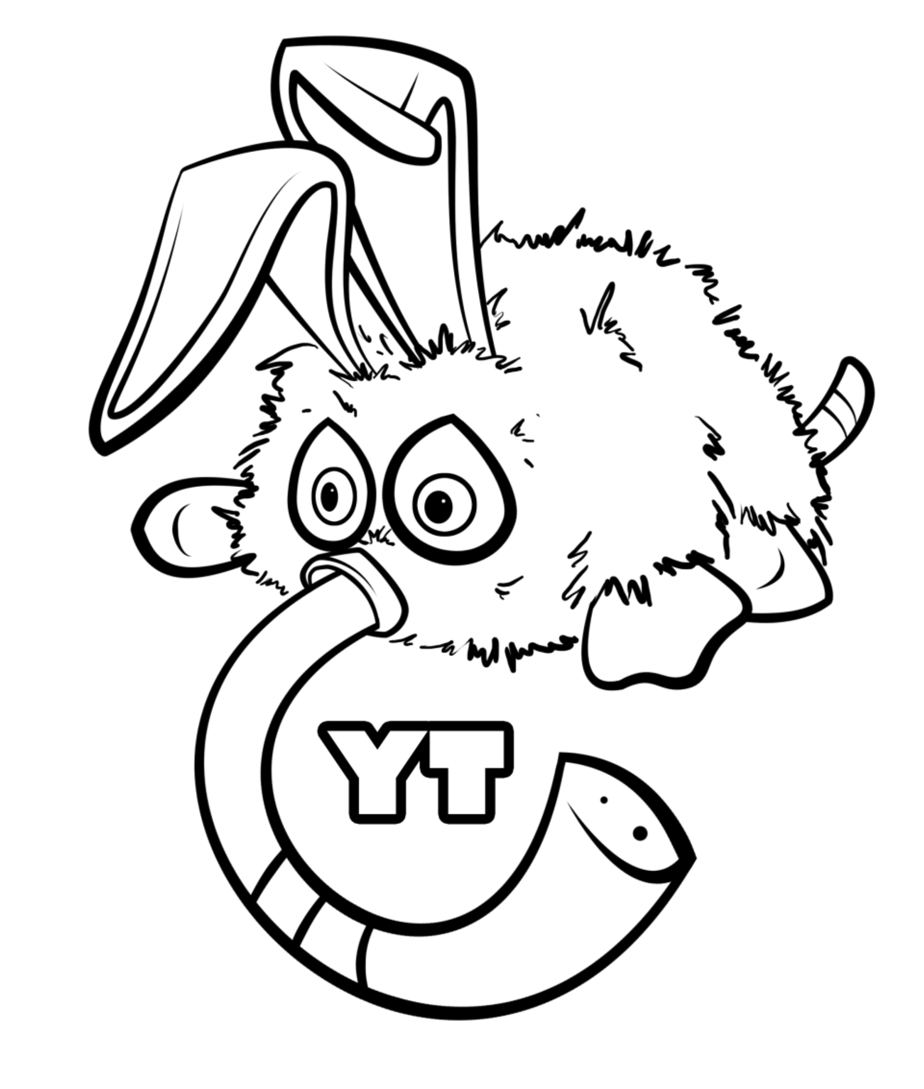
- Data Storage
- Scheduler
- Map Reduce

HL HighLoad++ Весна 2021

YT

- Миллион ядер
- Эксбибайт (EiB) диска
- Около 10 тысяч пользователей

- И это только на одном кластере!



Spark + YT

- Запустить на ресурсах кластера YT
- Читать данные из YT
- Писать данные в YT

- Показать wall time и cpu time лучше, чем у существующих решений

HL HighLoad++ Весна 2021

YT – файловая система

HDFS

```
/
|---> tmp
      |---> file.txt
      |---> example
            |---> dataset_1
                  |-----> part-0.parquet
                  |-----> part-1.parquet
                  |-----> part-2.parquet
            |---> dataset_2
                  |-----> part-0.parquet
                  |-----> part-1.parquet
```

YT

```
/
|---> tmp
      |---> file.txt
      |---> example
            |----> example_table_1
            |----> example_table_2
```


HL HighLoad++ Весна 2021

Таблица в YT

User level

#	a	b	c
0	"cat"	true	1
1	"dog"	true	20
2	"otter"	false	350

Attributes:

schema
a - String
b - Boolean
c - Integer

rowCount

Storage level

Columnar chunk 1
rows 0 - 200

Columnar chunk 2
rows 201 - 400

Columnar chunk 3
rows 401 - 600

Attributes:

schema
a - String
b - Boolean
c - Integer

rowCount

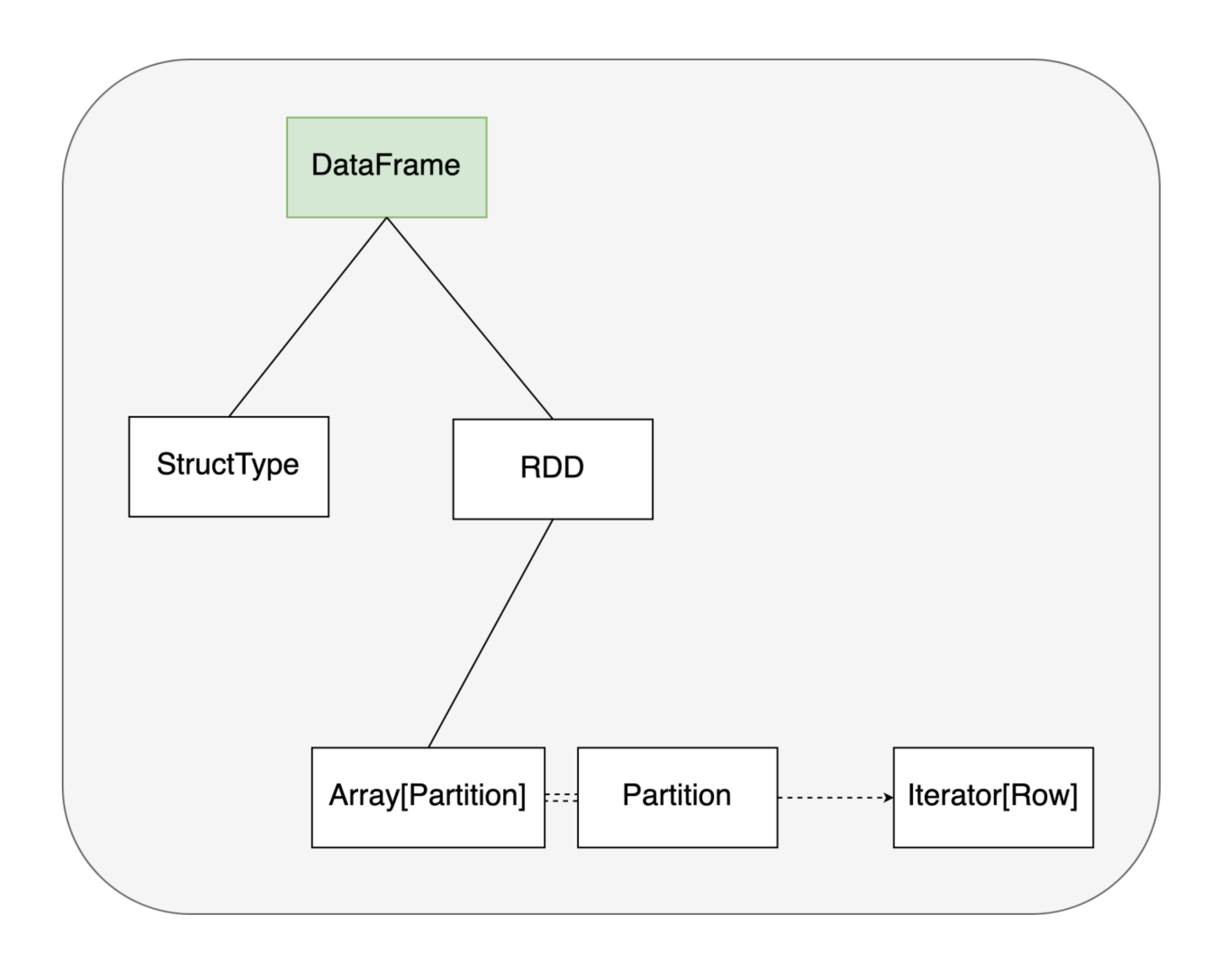
Простое решение: чтение

Дисклеймер

- API – набор классов / интерфейсов / методов, которые мы *вызываем*
- SPI – набор классов / интерфейсов / методов, которые мы *расширяем или реализуем*

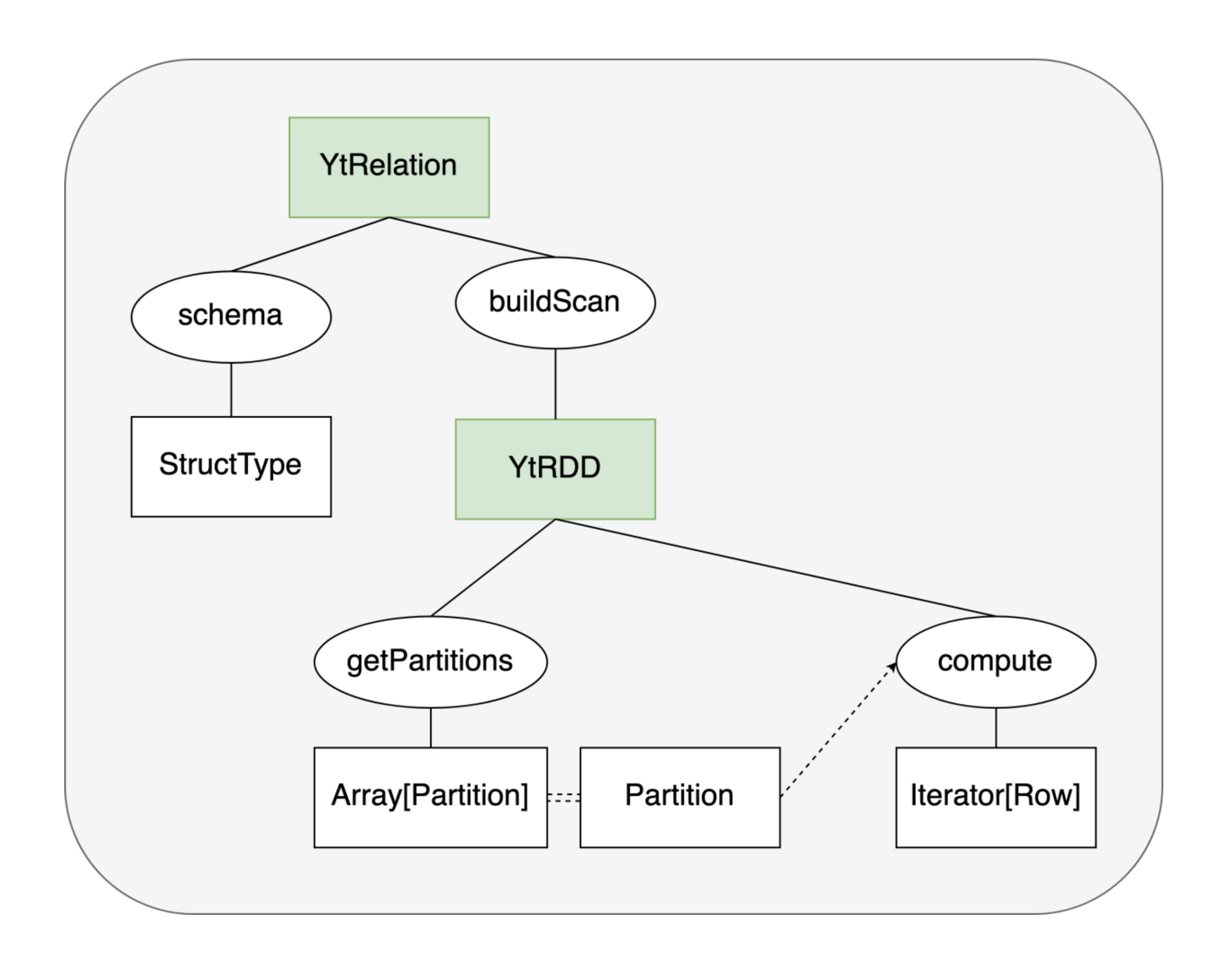
HL HighLoad++ Весна 2021

Spark SPI



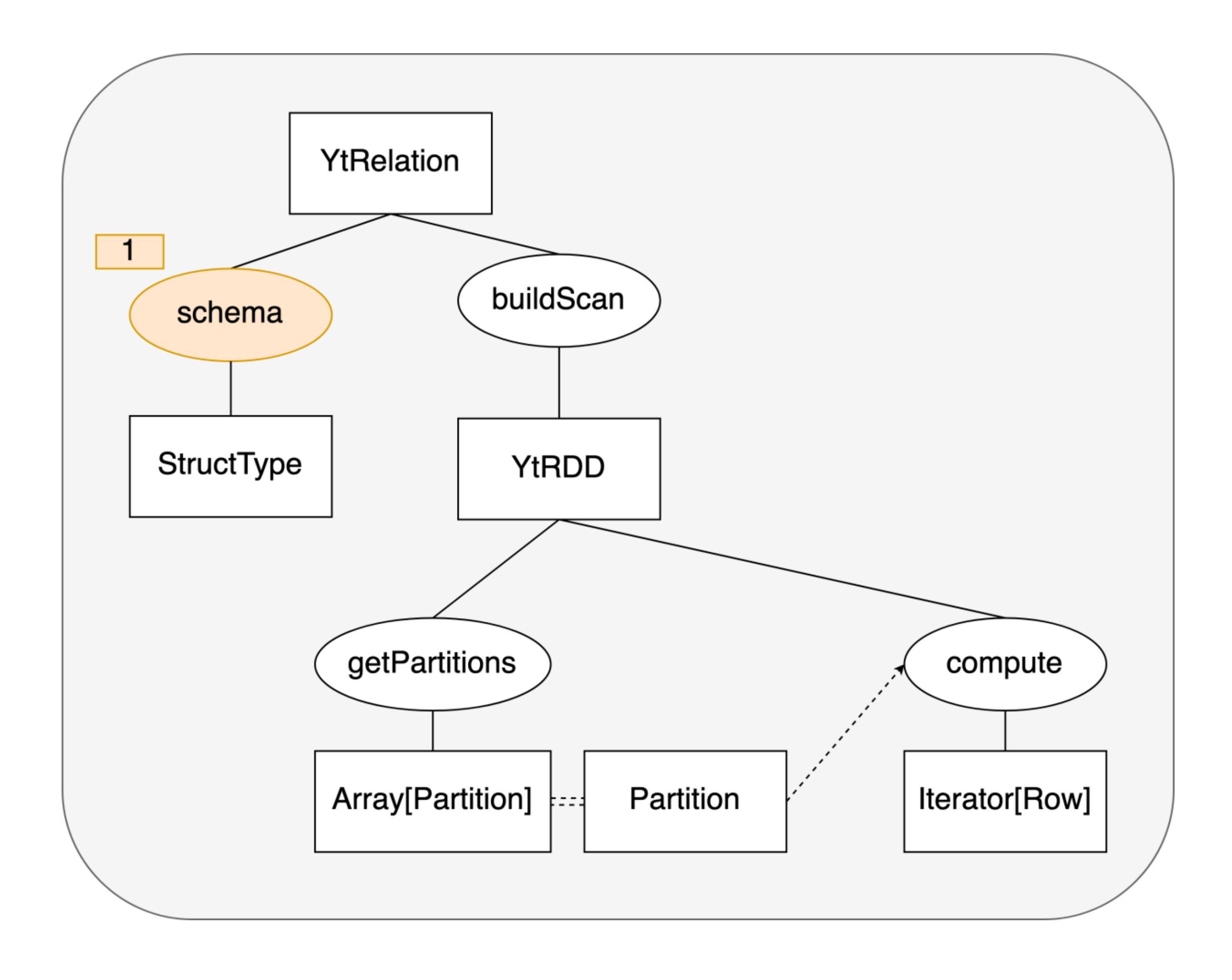
HL HighLoad++ Весна 2021

Spark SPI



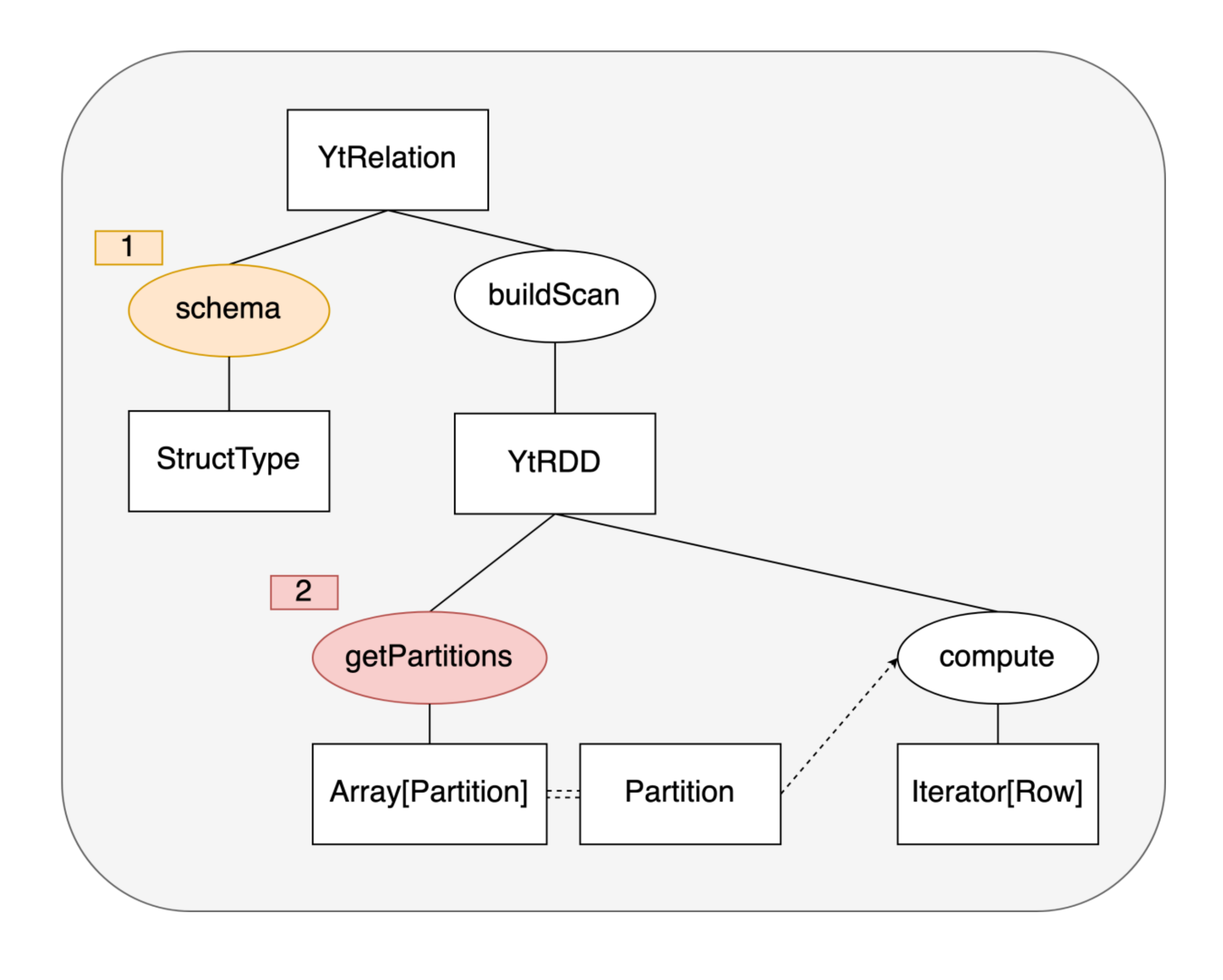
HL HighLoad++ Весна 2021

Получение схемы



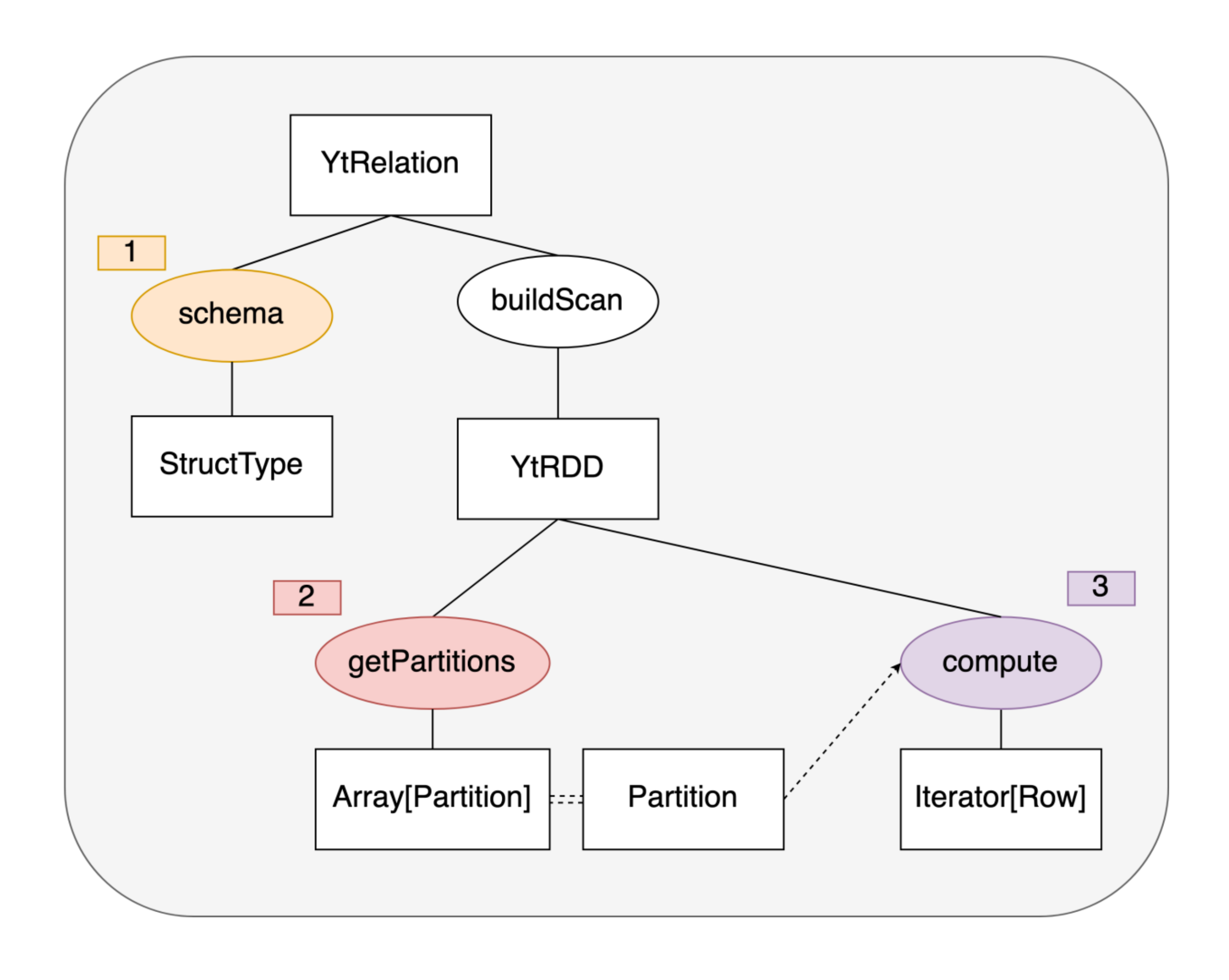
HL HighLoad++ Весна 2021

Получение партиций



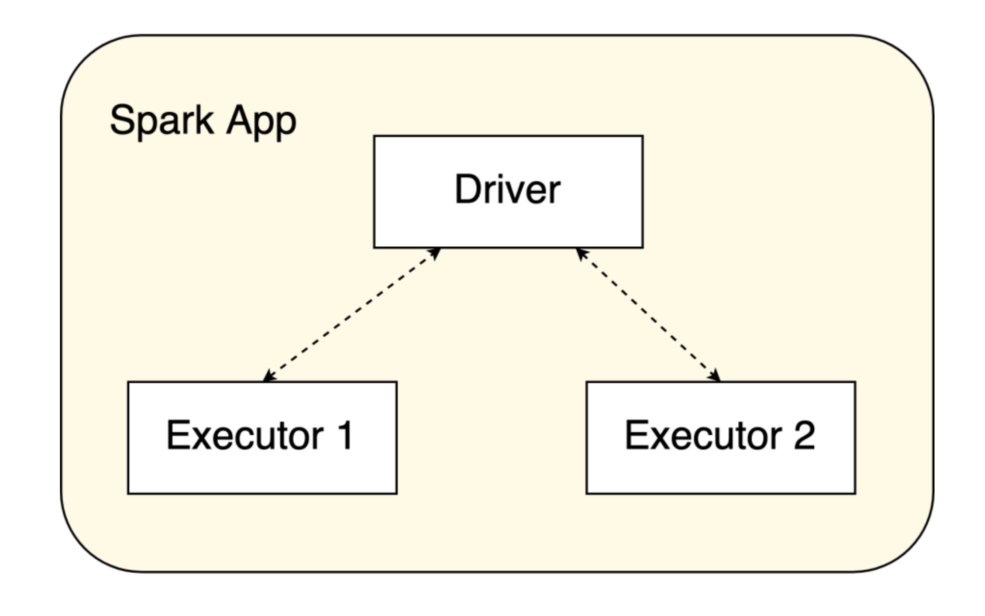
HL HighLoad++ Весна 2021

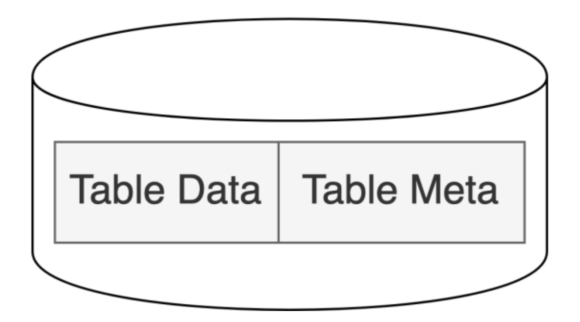
Чтение партиции



HL HighLoad++ Весна 2021

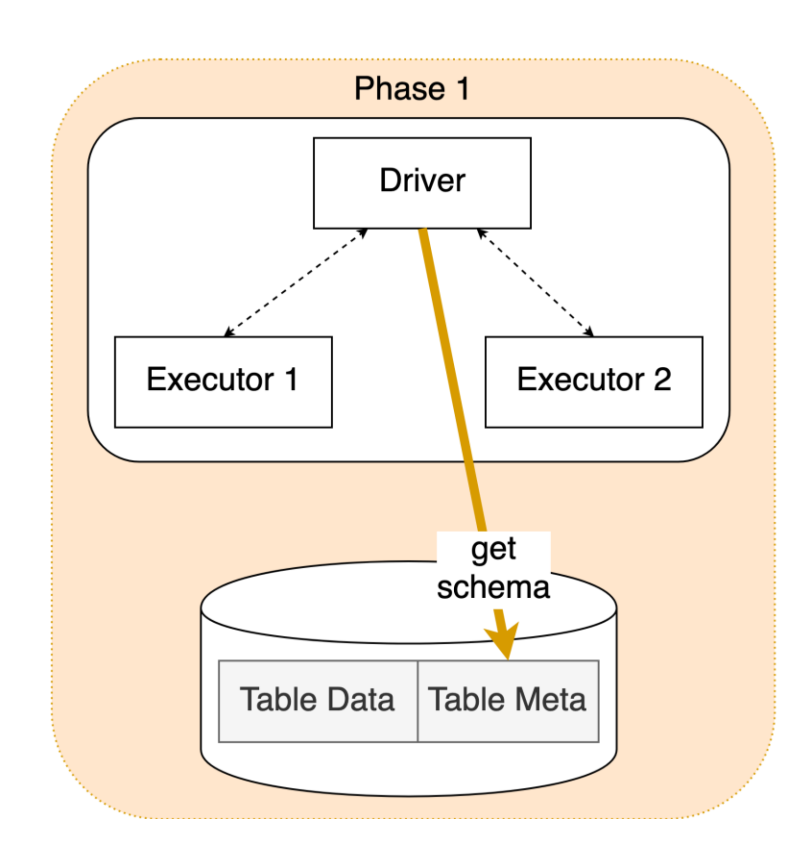
Как это выполняется





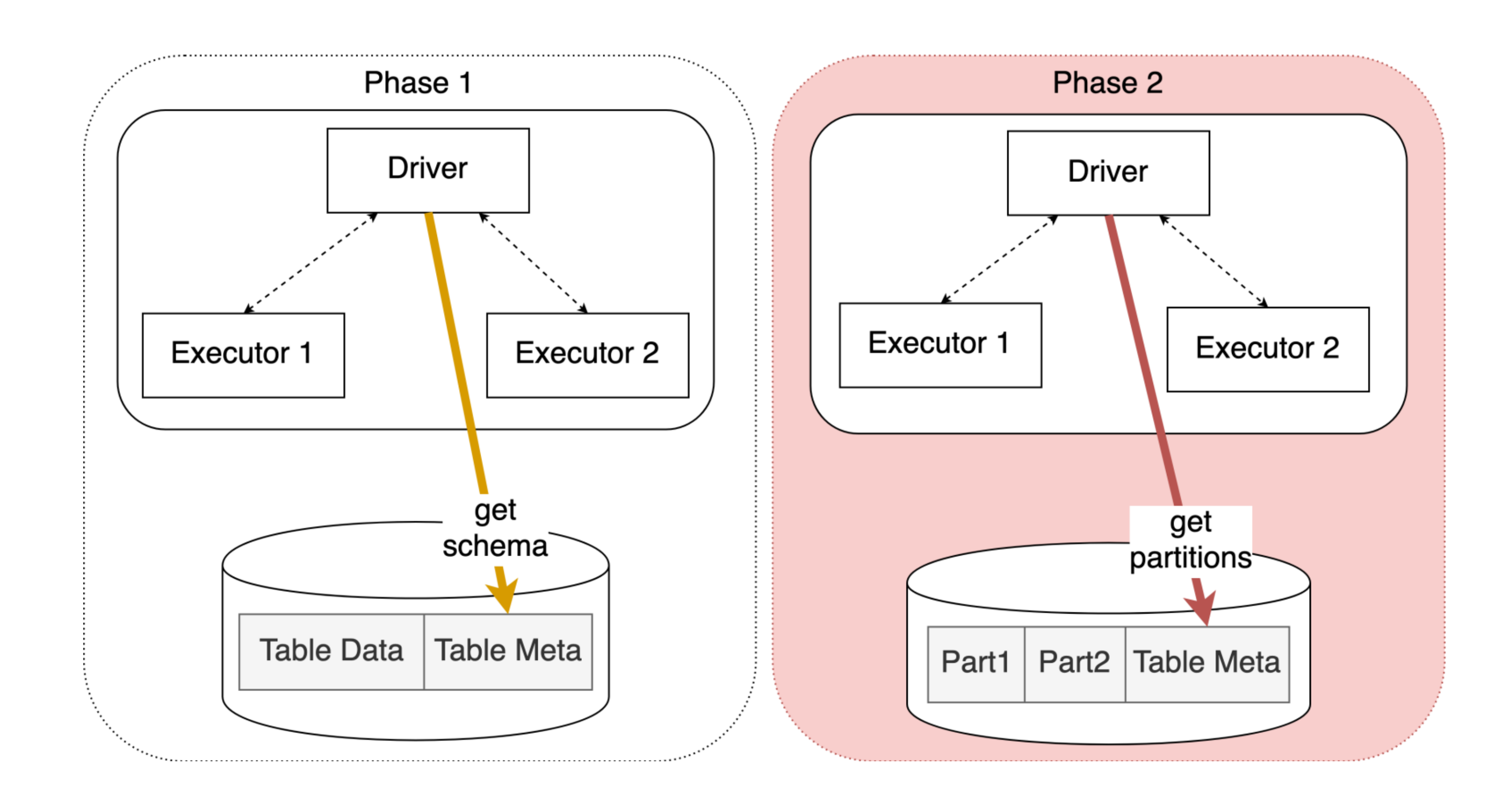
HL HighLoad++ Весна 2021

Получение схемы



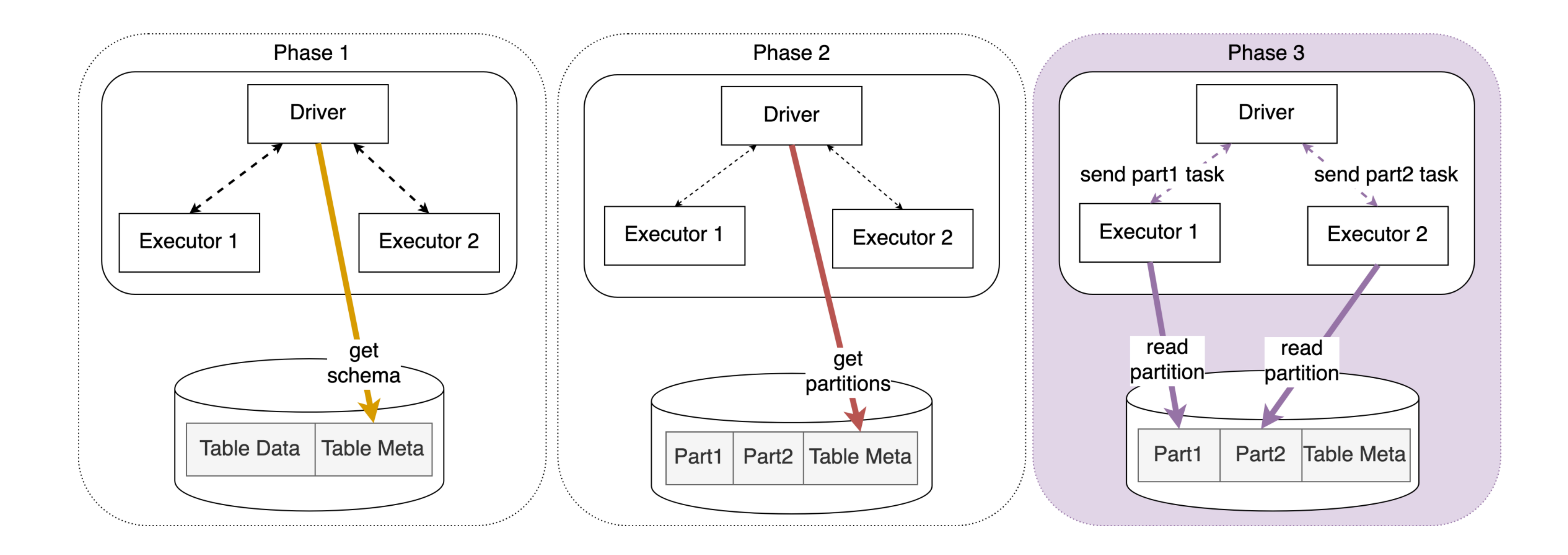
HL HighLoad++ Весна 2021

Получение партиций



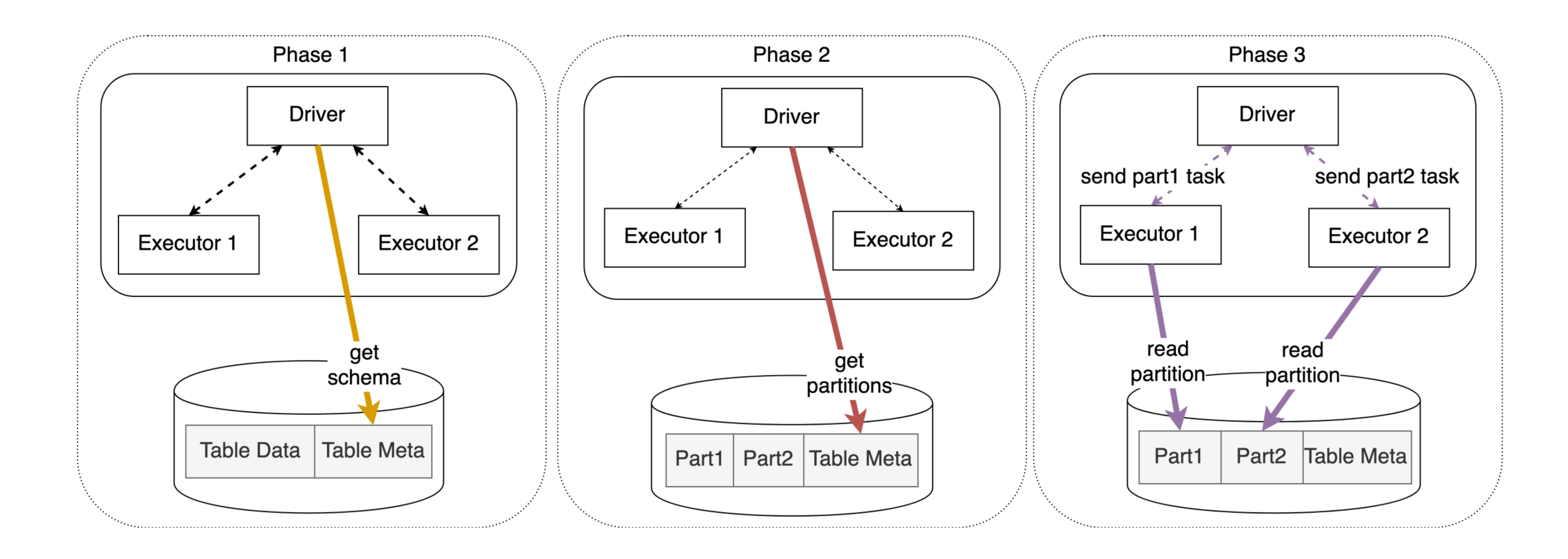
HL HighLoad++ Весна 2021

Чтение партиции



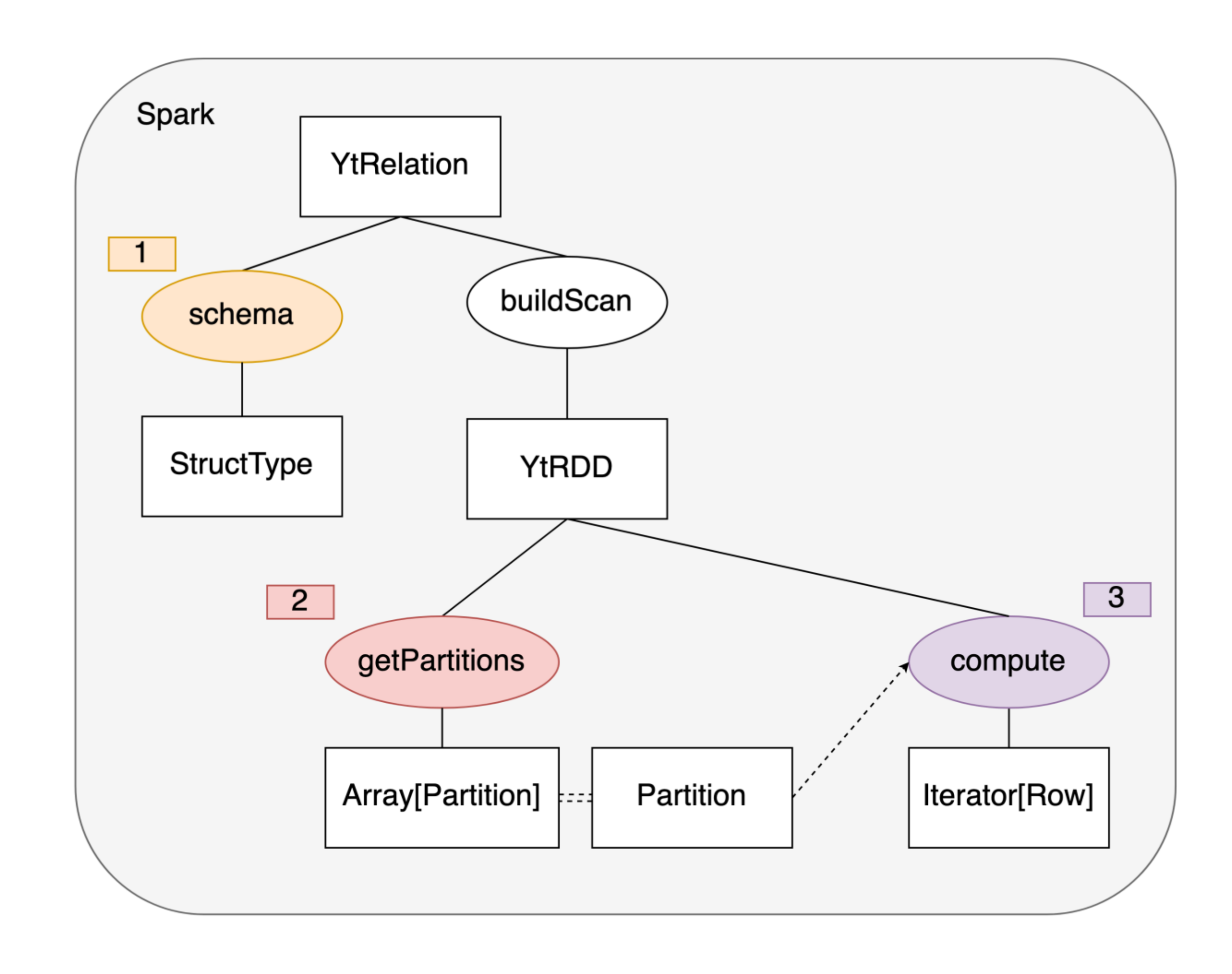
HL HighLoad++ Весна 2021

Как это выполняется



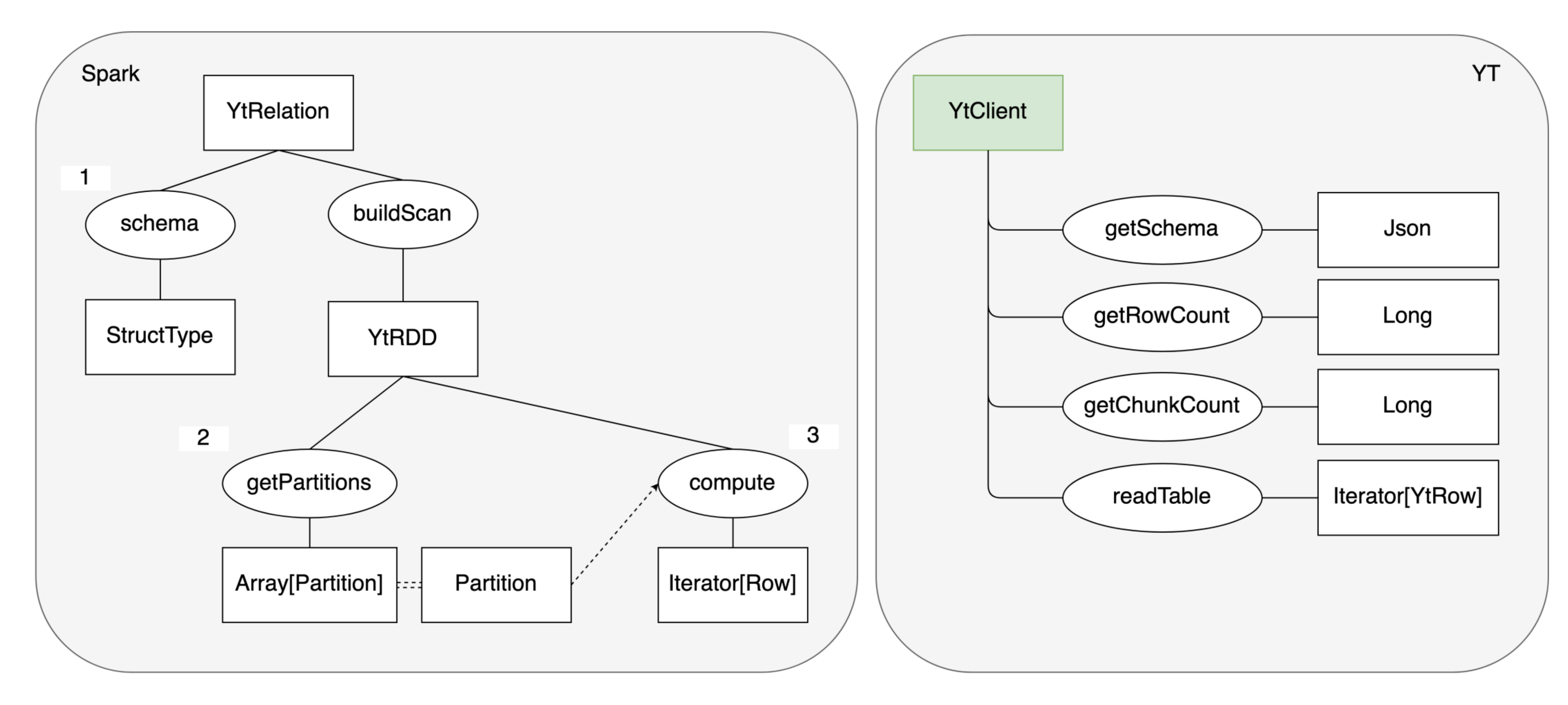
HL HighLoad++ Весна 2021

Как это выглядит в коде



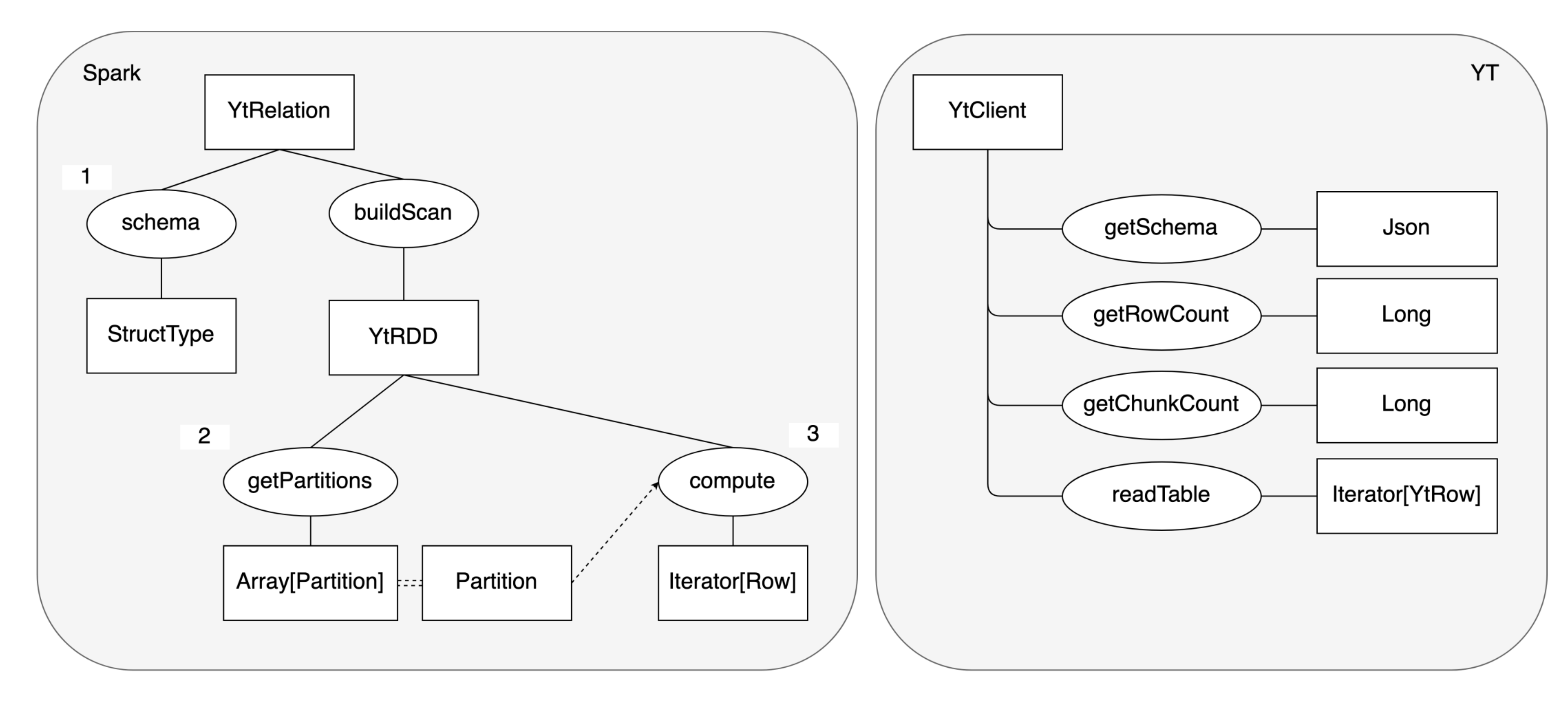
HL HighLoad++ Весна 2021

YT API



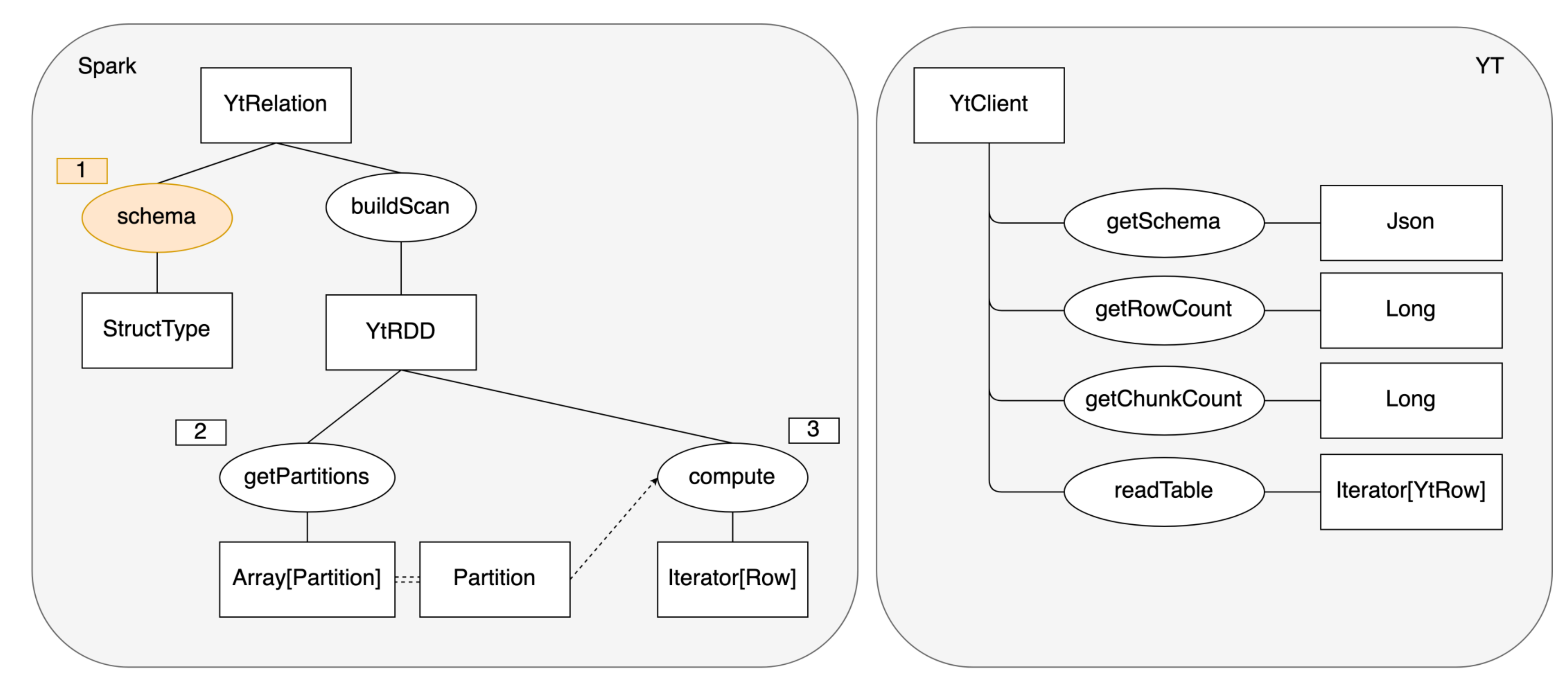
HL HighLoad++ Весна 2021

SPYT



HL HighLoad++ Весна 2021

Получение схемы



HL HighLoad++ Весна 2021

Преобразование схемы в StructType

```
{
    "a": "int32",
    "b": "string",
    "c": "uint64",
    "d": "any",
    "name.with.dots": "double",
}
```

→ StructType

HL HighLoad++ Весна 2021

Простые типы

```
{
  "a": "int32",
  "b": "string",
  "c": "uint64",
  "d": "any",
  "name.with.dots": "double",
}
```

```
StructType (
  SF("a", IntegerType),
  SF("b", StringType),

)
```


HL HighLoad++ Весна 2021

Дополнительные типы

```
{
   "a": "int32",
   "b": "string",
   "c": "uint64",
   "d": "any",
   "name.with.dots": "double",
}
```

```
StructType (
   SF("a", IntegerType),
   SF("b", StringType),
   SF("c", StringType,
      +original type in meta)

)
```

Подсказки для типов

```
{
    "a": "int32",
    "b": "string",
    "c": "uint64",
    "d": "any",
    "name.with.dots": "double",
}

+ schema_hint
"d": ArrayType(StringType)
```

```
StructType (
    SF("a", IntegerType),
    SF("b", StringType),
    SF("c", StringType, +meta)
    SF("d", ArrayType(StringType)),


)
```


HL HighLoad++ Весна 2021

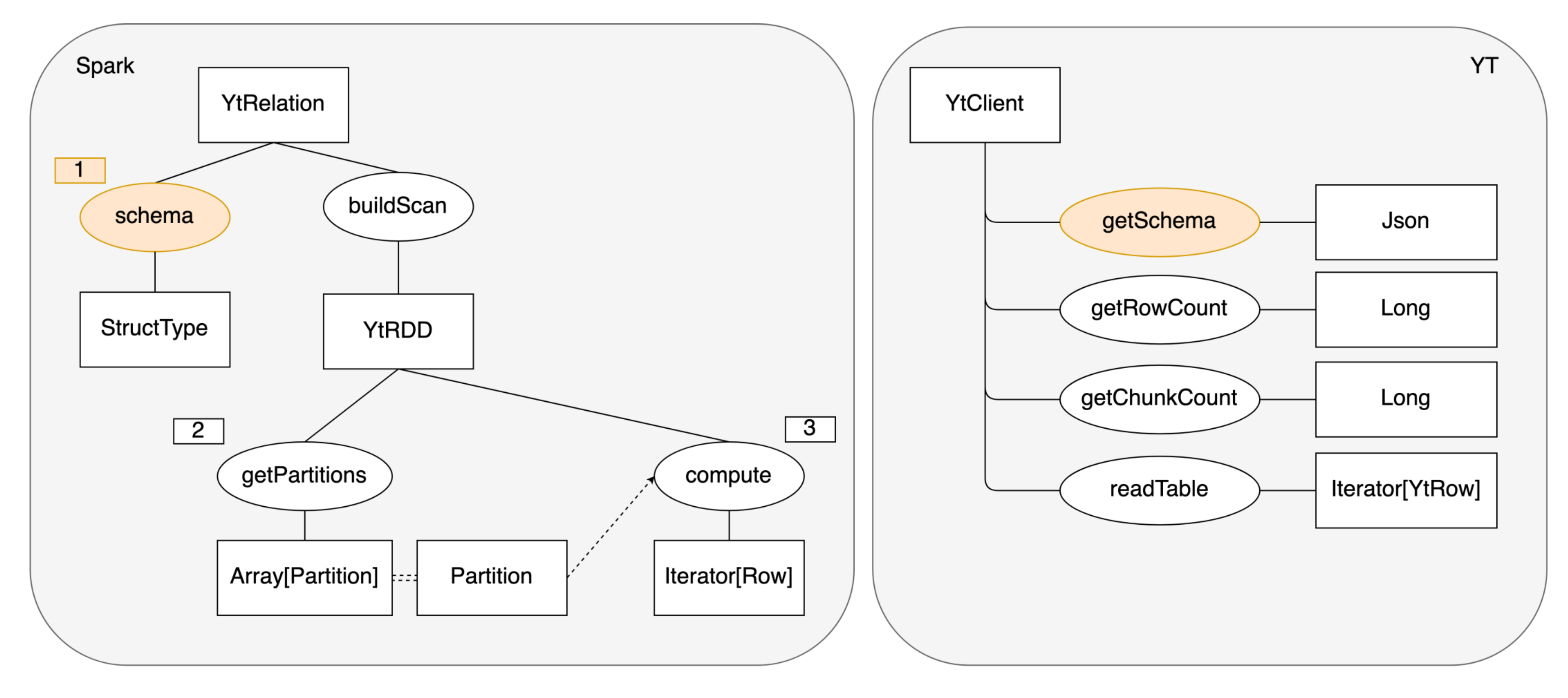
Переименование

```
{
  "a": "int32",
  "b": "string",
  "c": "uint64",
  "d": "any",
  "name.with.dots": "double",
}
```

```
StructType (
  SF("a", IntegerType),
  SF("b", StringType),
  SF("c", StringType, +meta)
  SF("d", ArrayType(StringType)),
  SF("name_with_dots",
      DoubleType,
      + original name in meta)
)
```

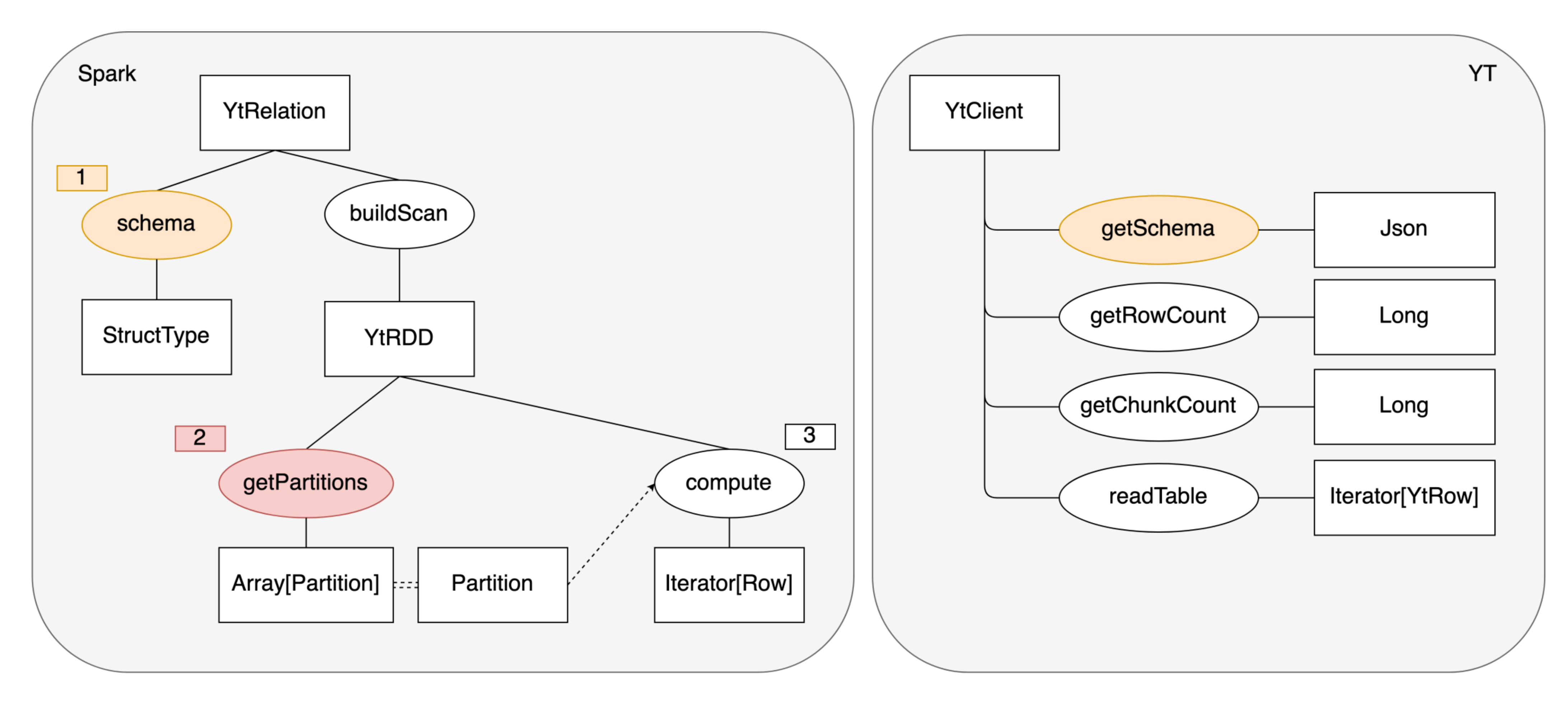

HL HighLoad++ Весна 2021

Получение схемы



HL HighLoad++ Весна 2021

Получение партиций



HL HighLoad++ Весна 2021

Таблица в YT

User level

#	a	b	c
0	"cat"	true	1
1	"dog"	true	20
2	"otter"	false	350

Attributes:

schema
a - String
b - Boolean
c - Integer

rowCount

Storage level

Columnar chunk 1
rows 0 - 200

Columnar chunk 2
rows 201 - 400

Columnar chunk 3
rows 401 - 600

Attributes:

schema
a - String
b - Boolean
c - Integer

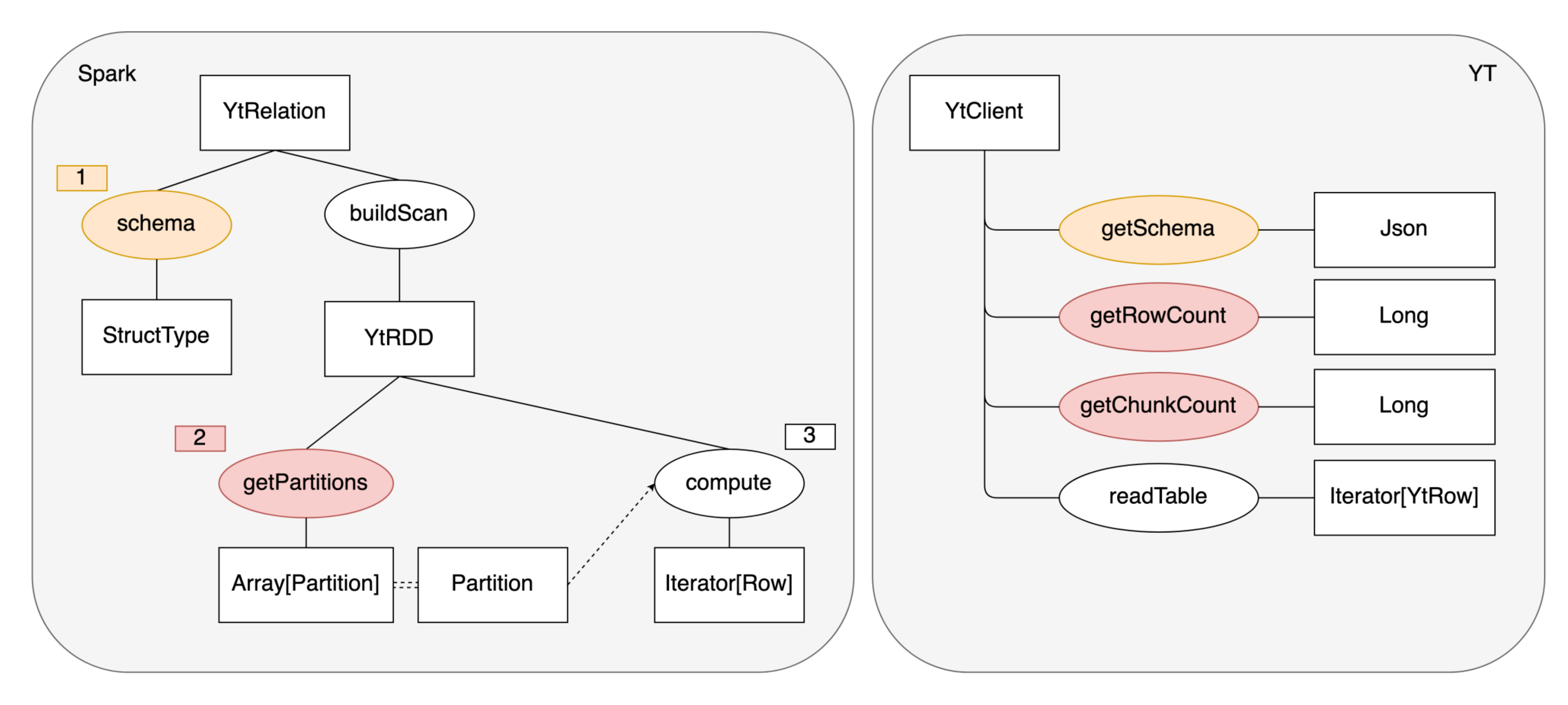
rowCount

Разбиение таблицы на партиции

- Можем узнать точное количество строк в таблице из метаданных
- Можем запрашивать диапазоны строк при чтении

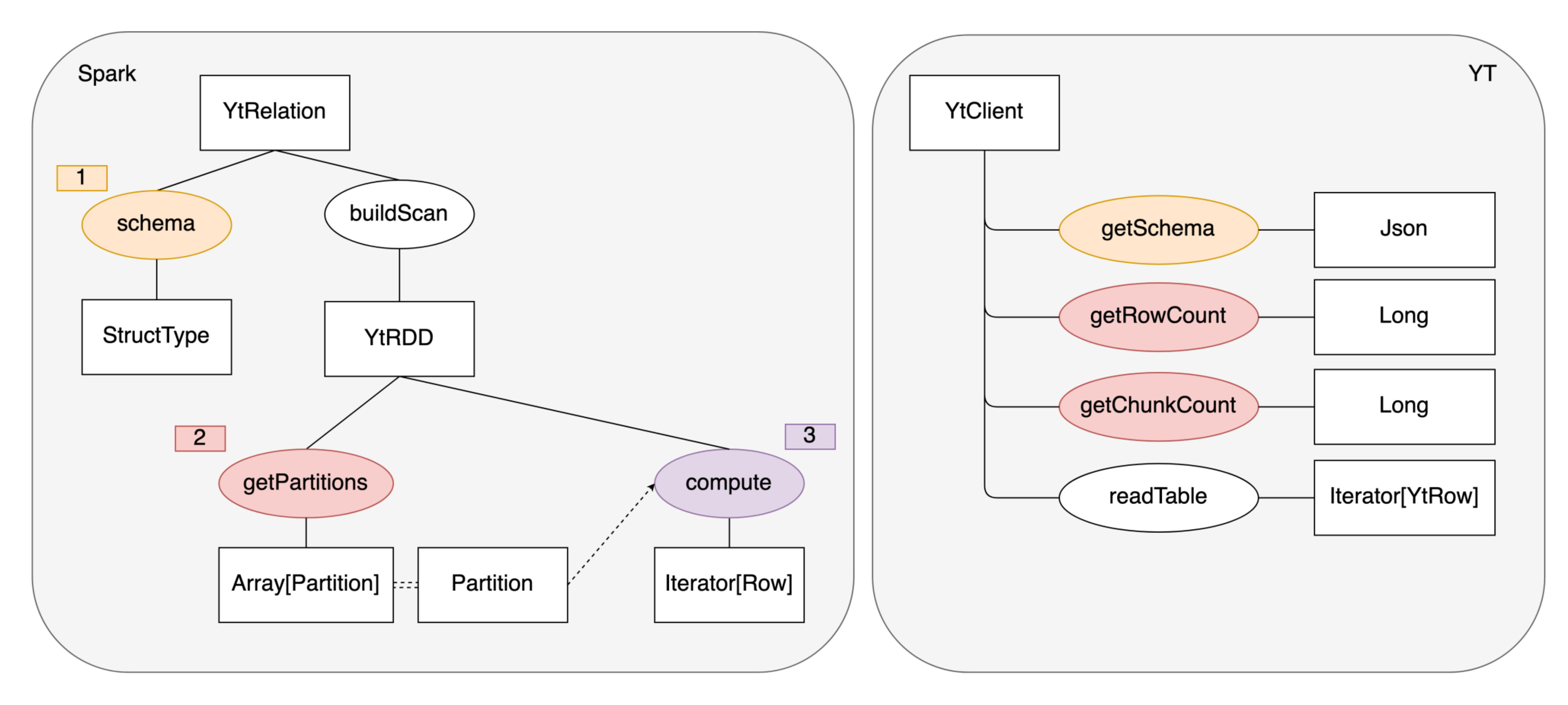
- Бывают проблемы с равномерностью распределения

Получение партиций



HL HighLoad++ Весна 2021

Чтение партиции



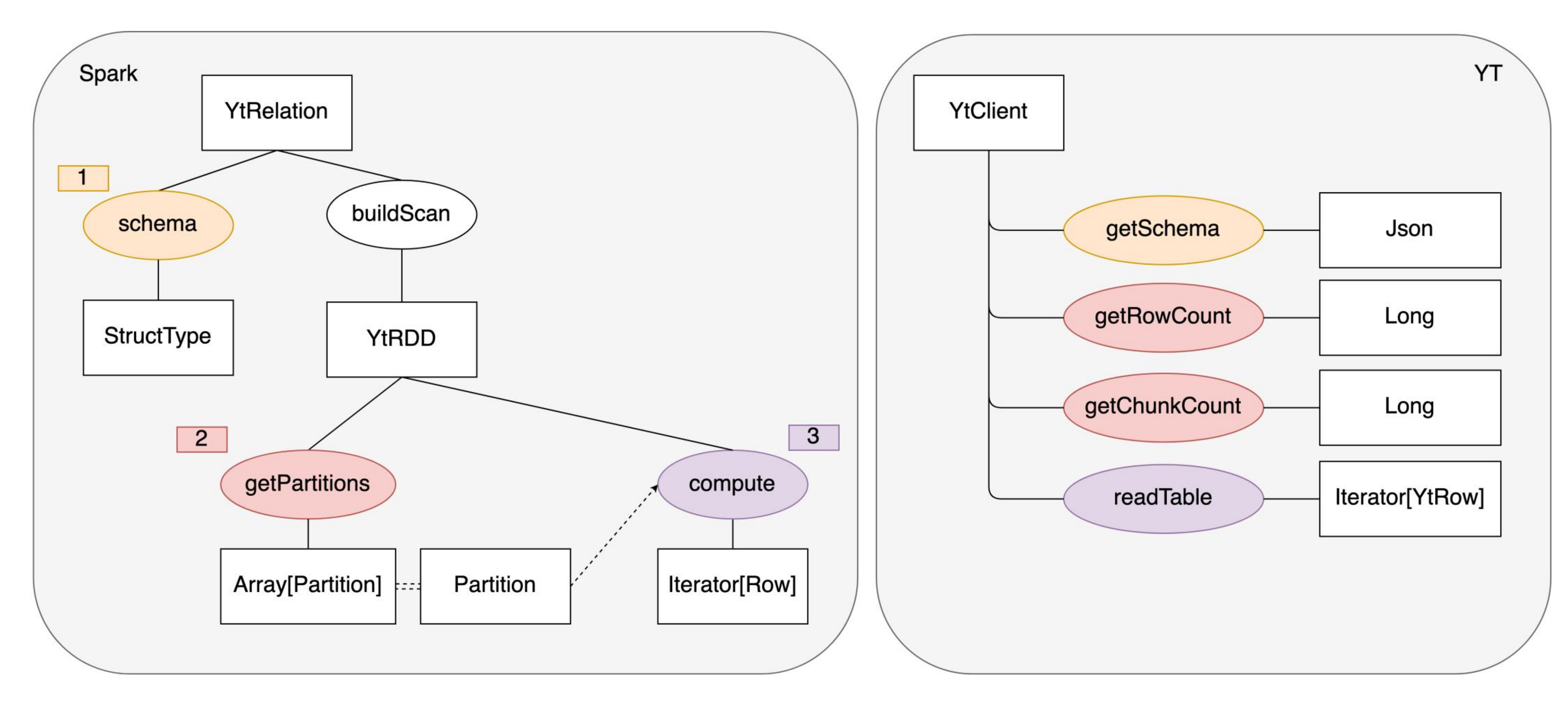
HL HighLoad++ Весна 2021

Чтение партиции из YT

- API отдаёт строки таблицы в своём формате YtRow
- Нужно преобразовывать строки в спарковый Row
- В этот момент нужно сделать преобразование значений под спарковую систему типов

HL HighLoad++ Весна 2021

Чтение

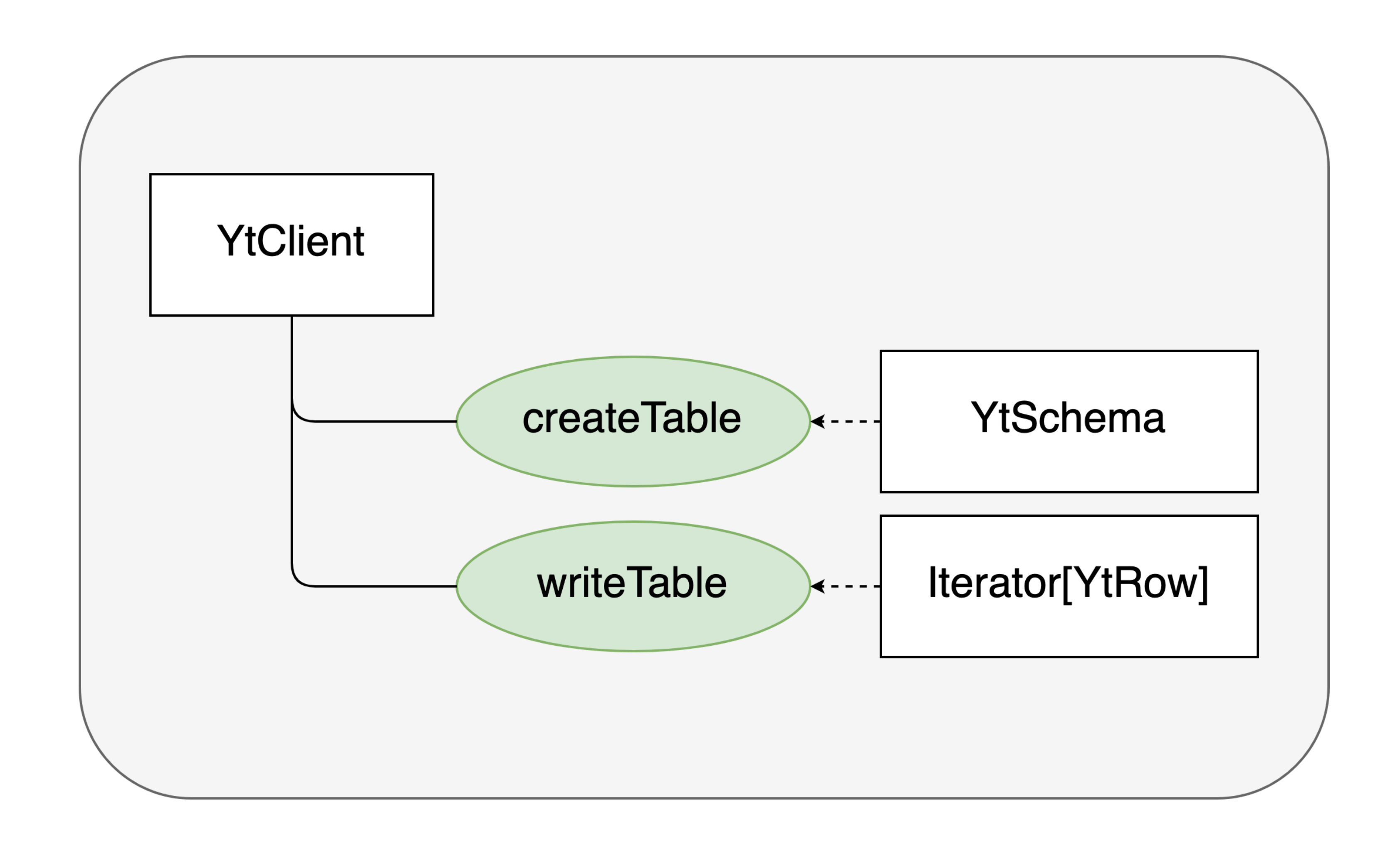


HL HighLoad++ Весна 2021

Простое решение: запись

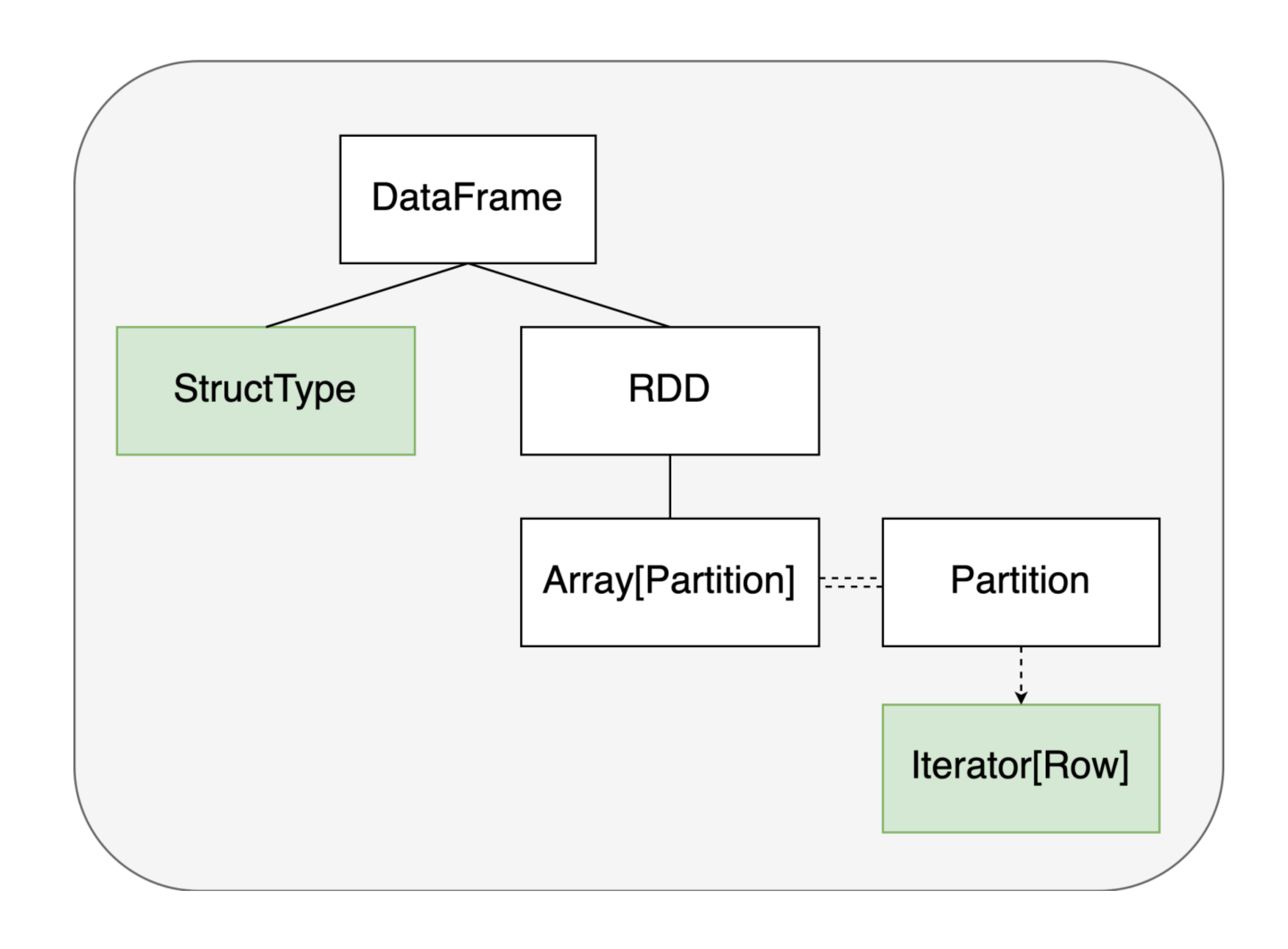
HL HighLoad++ Весна 2021

YT API



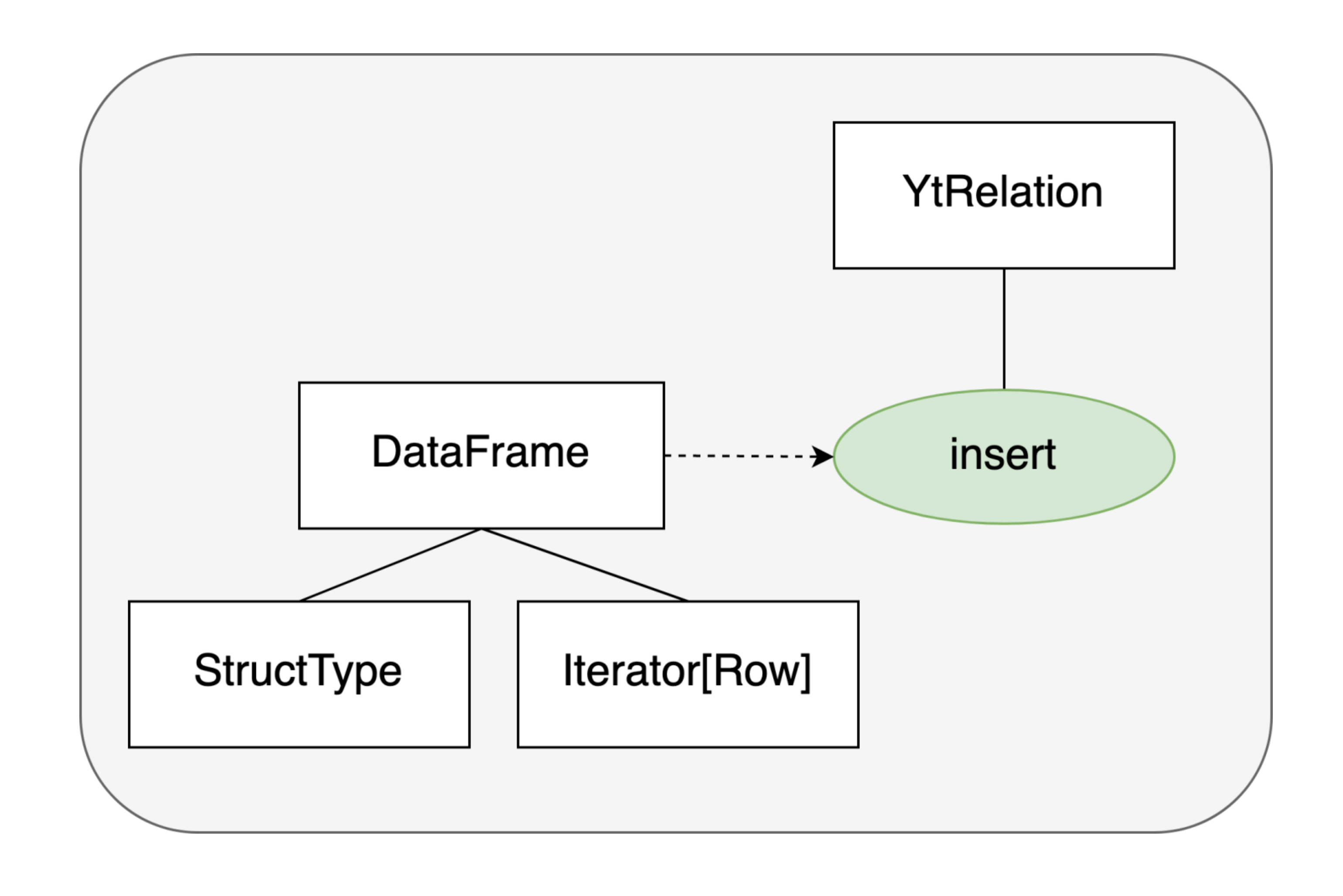
HL HighLoad++ Весна 2021

Spark SPI



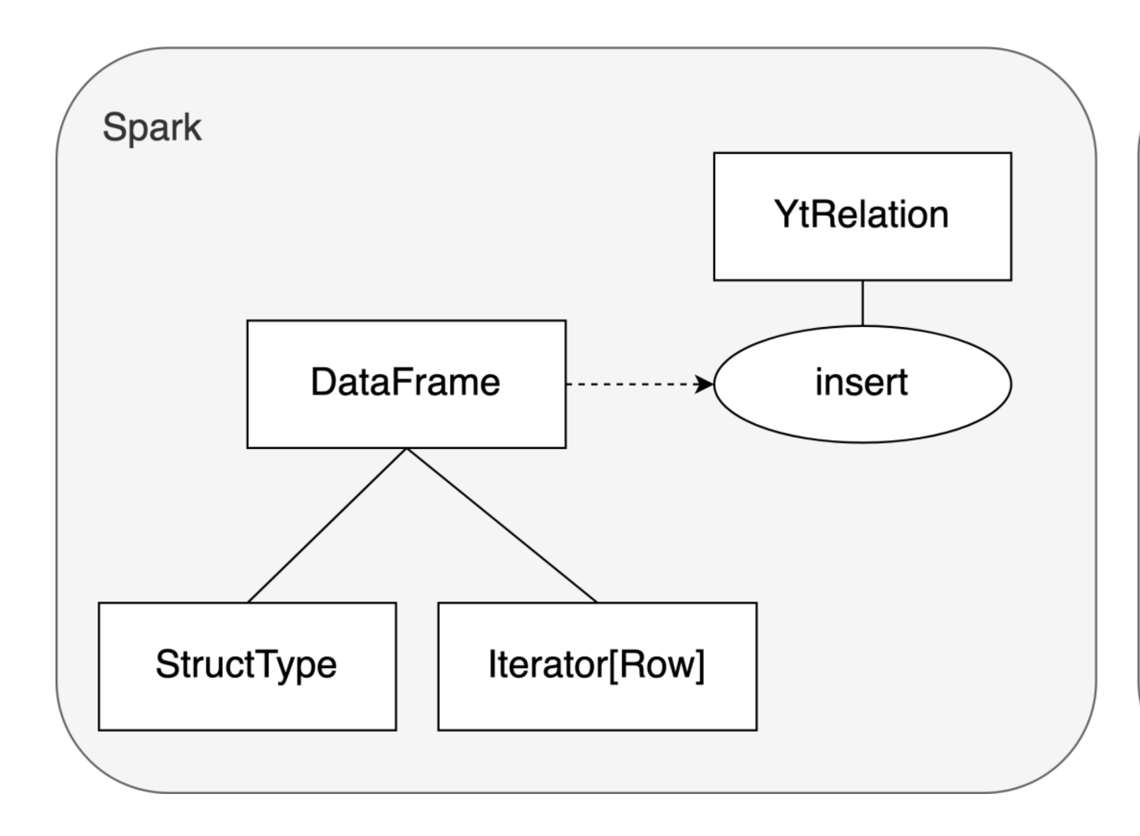
HL HighLoad++ Весна 2021

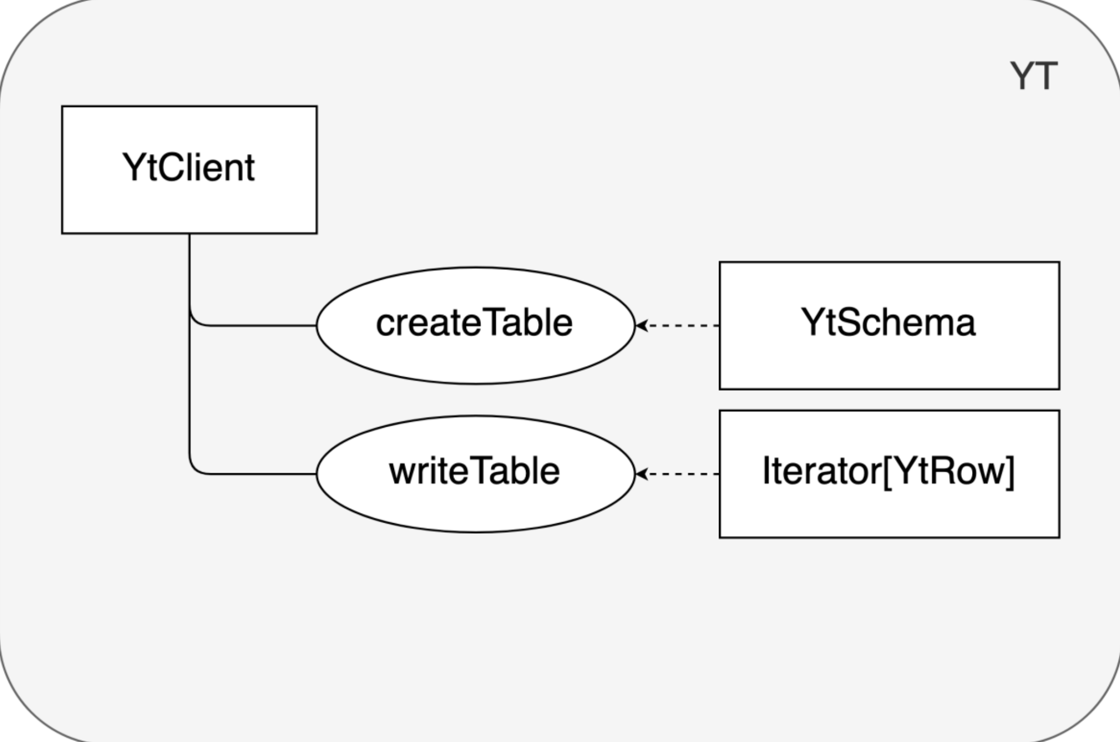
Spark SPI



HL HighLoad++ Весна 2021

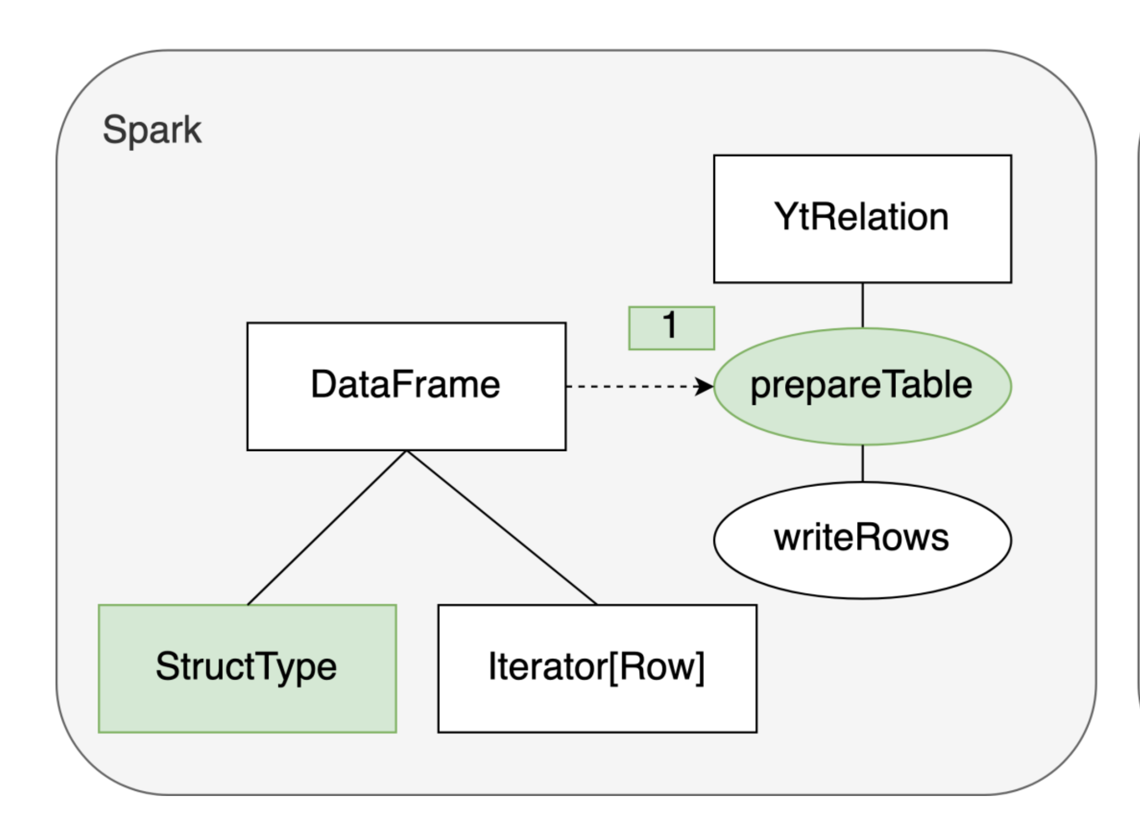
SPYT

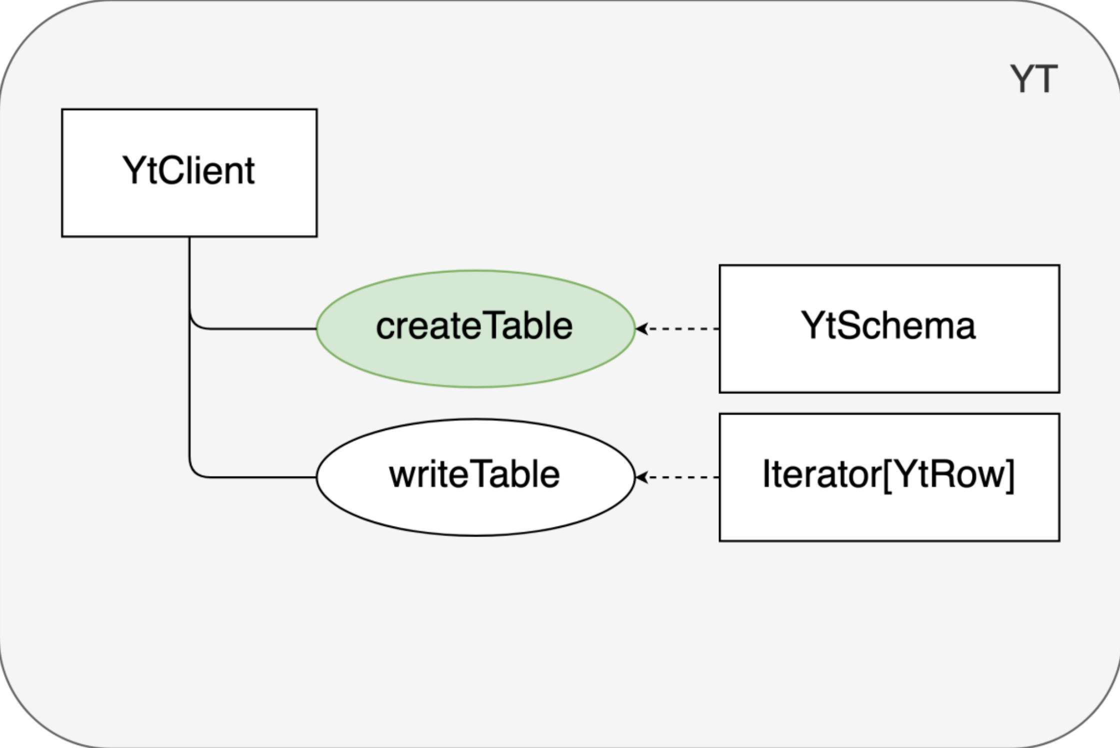




HL HighLoad++ Весна 2021

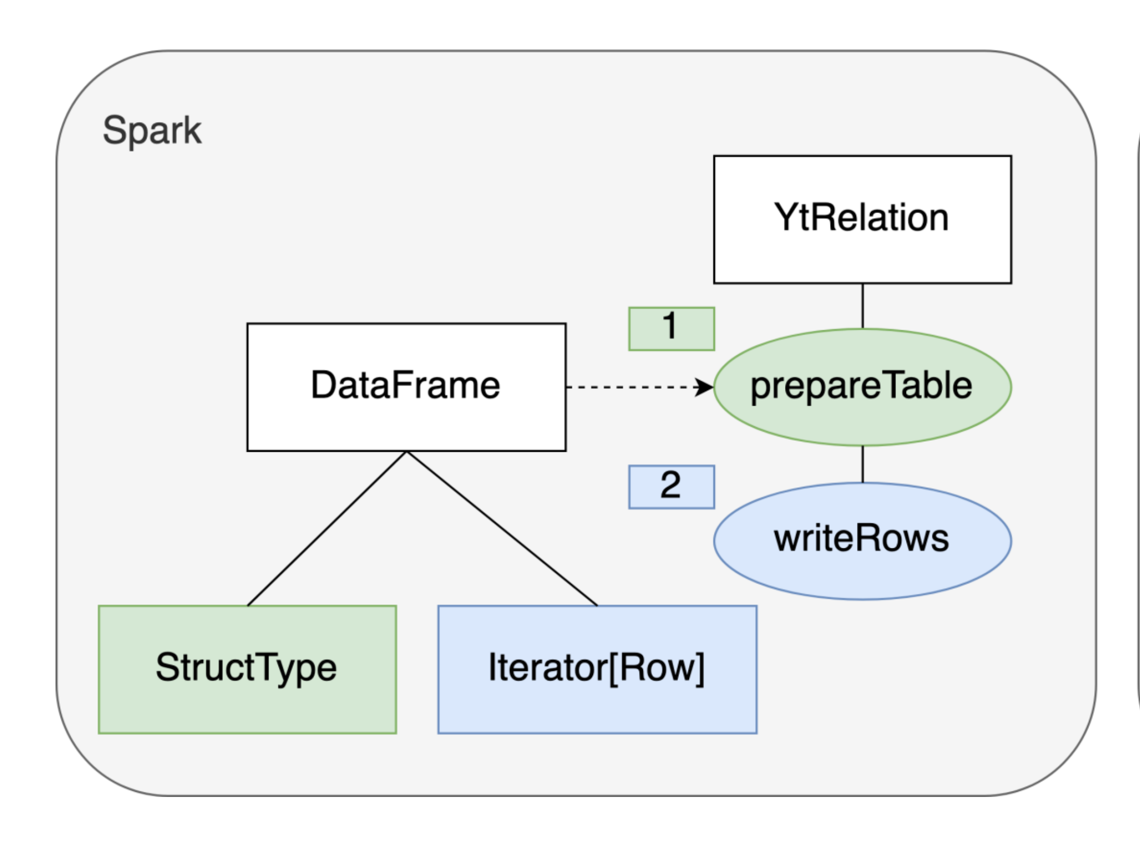
Подготовка записи

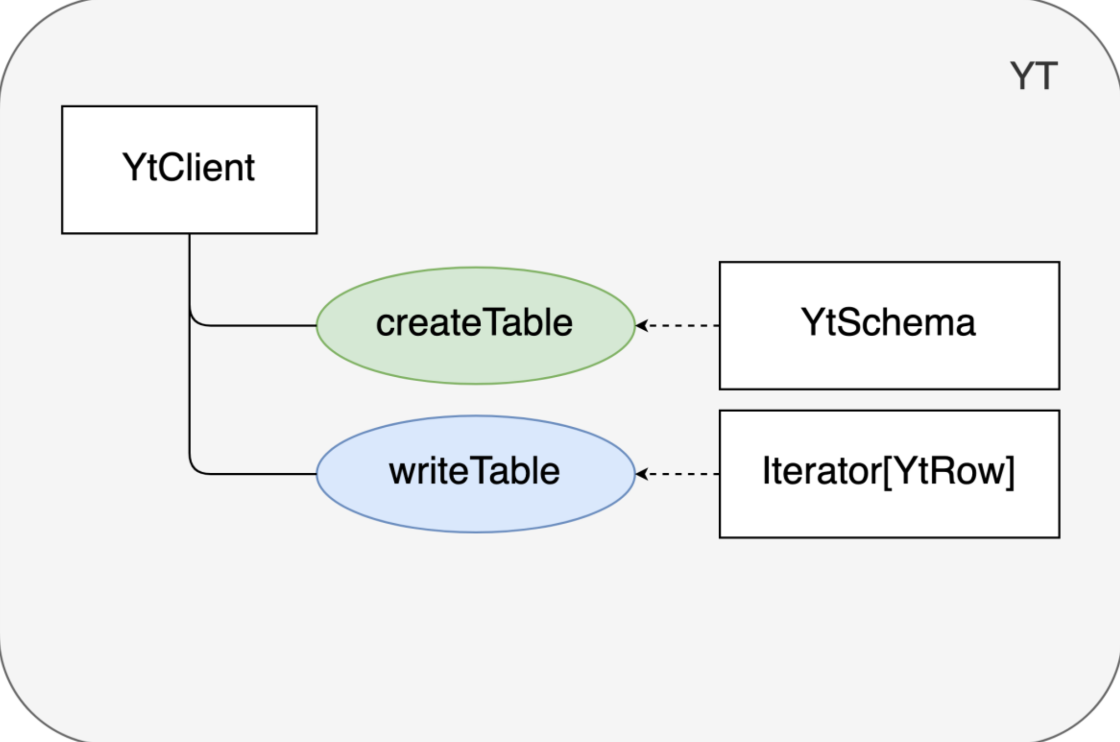




HL HighLoad++ Весна 2021

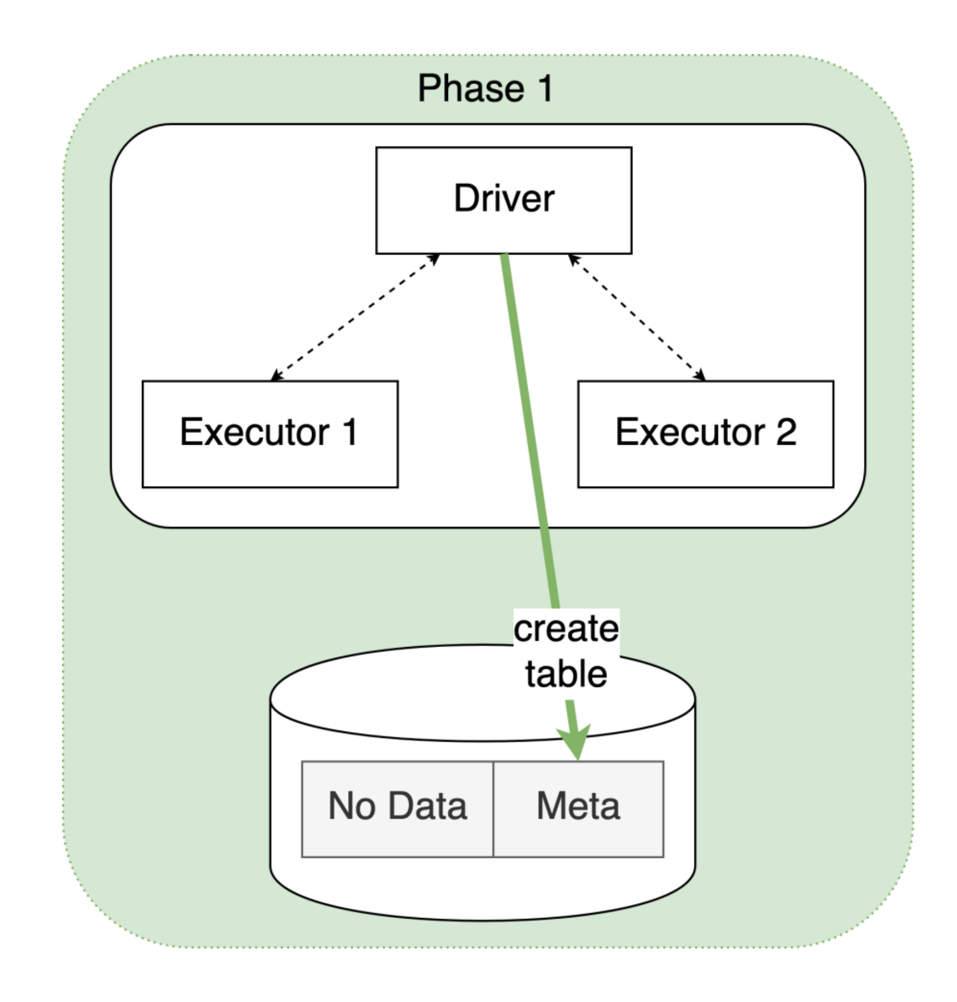
Запись партиций





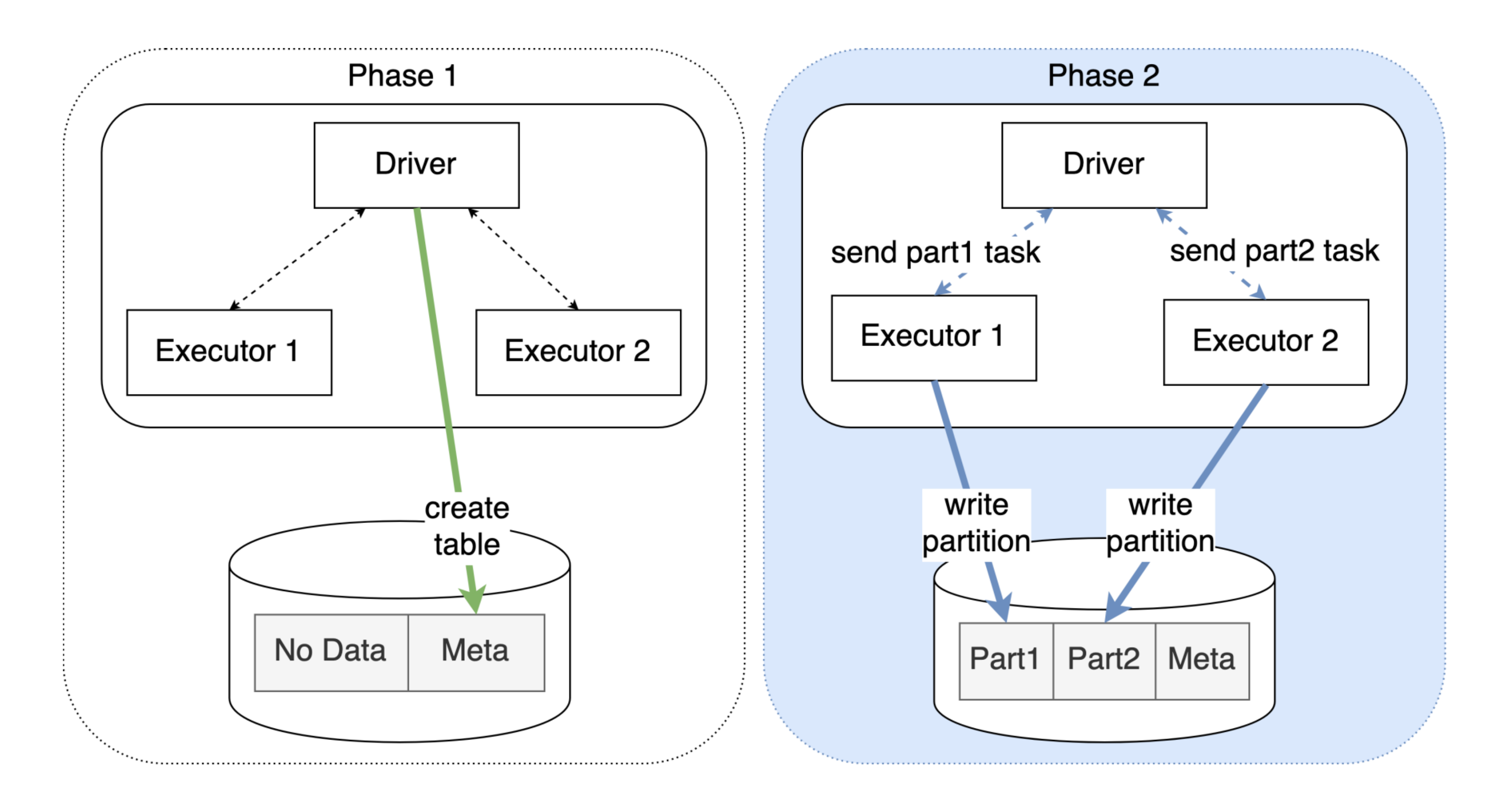
HL HighLoad++ Весна 2021

Подготовка таблицы



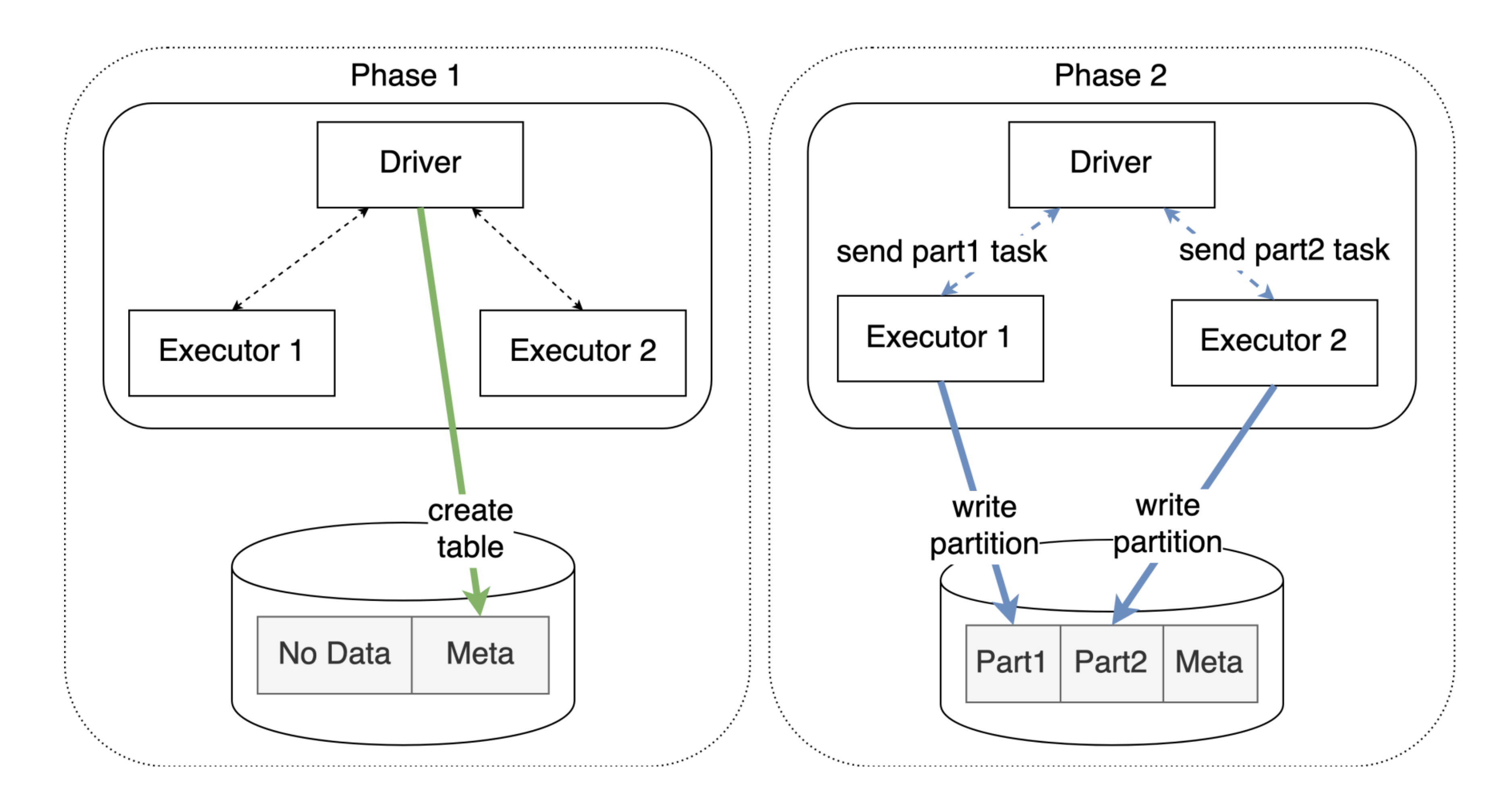
HL HighLoad++
Весна 2021

Запись партиций

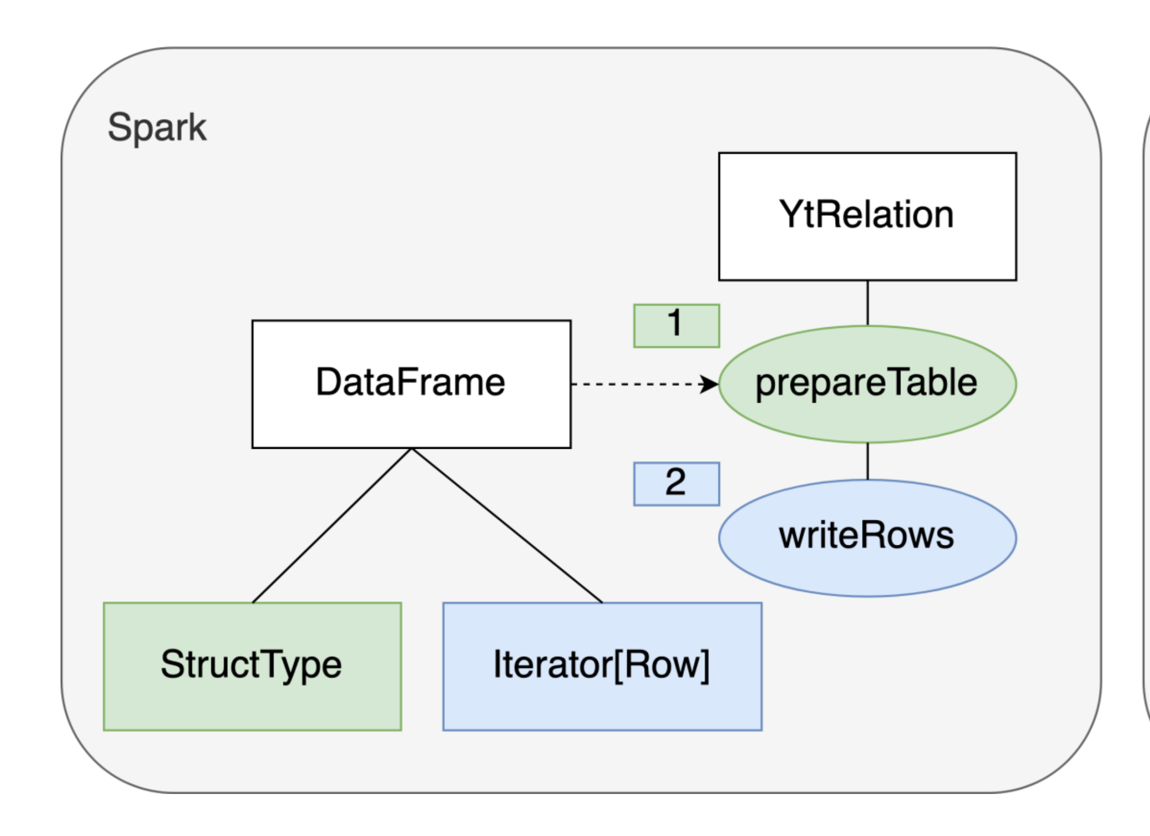


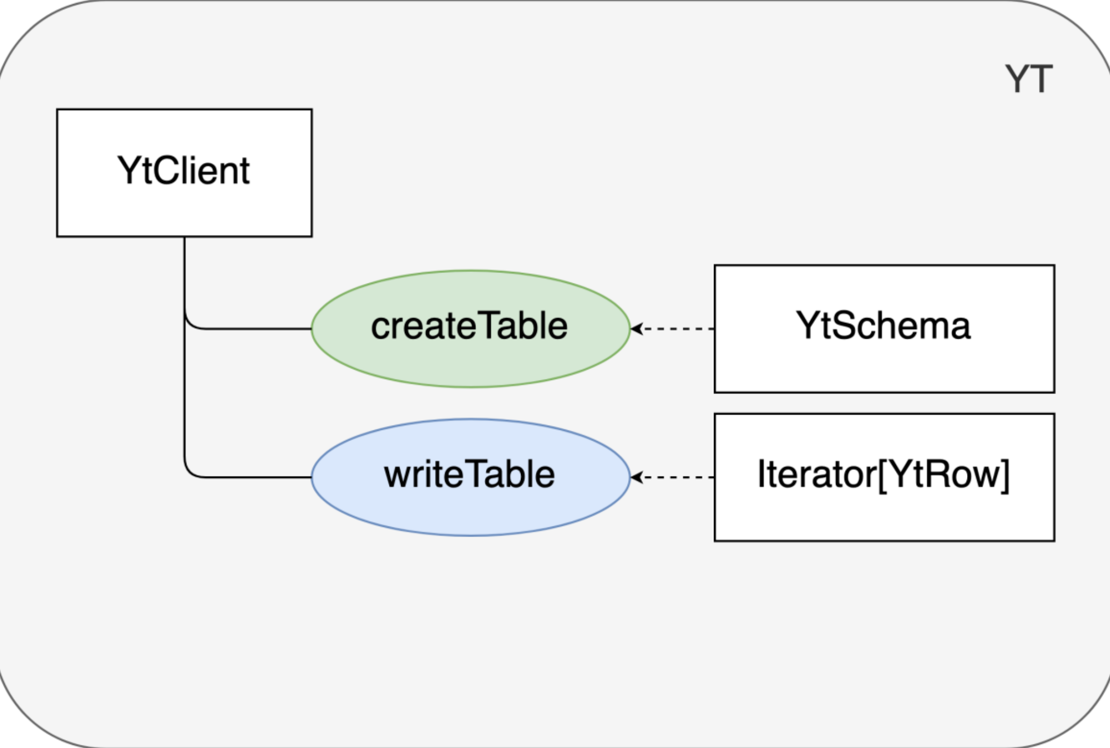
HL HighLoad++ Весна 2021

Как это выполняется



Как это выглядит в коде





HL HighLoad++ Весна 2021

Особенности записи

- Обратное преобразование схемы
- Обратное преобразование строчек таблицы
- Метод writeRows будет вызываться параллельно с разных экзекьюторов

HL HighLoad++ Весна 2021

Итог простого решения

- Чтение:
  - Запрос схемы с драйвера
  - Разделение на партиции на драйвере
  - Чтение партиции на экзекьюторе
- Запись:
  - Создание таблицы на драйвере
  - Запись партиций с экзекьюторов параллельно

Подробное выступление Jacek Laskowski: https://www.youtube.com/watch?v=vfd83ELlMfc

Что не так с простым подходом?

Что не так с простым подходом

- Нет чтения батчами

Таблица в YT

User level

#	a	b	c
0	"cat"	true	1
1	"dog"	true	20
2	"otter"	false	350

Attributes:

schema
a - String
b - Boolean
c - Integer

rowCount

Storage level

Columnar chunk 1 rows 0 - 200
Columnar chunk 2 rows 201 - 400
Columnar chunk 3 rows 401 - 600

Attributes:

schema
a - String
b - Boolean
c - Integer

rowCount

Батчи в YT

Storage level

Columnar chunk	Chunk meta
Columnar chunk 1 rows 0 - 200	Chunk meta 200 rows a - ["alpaca", "cat"]
Columnar chunk 2 rows 201 - 400	Chunk meta 200 rows a - ["dog","mongoose"]
Columnar chunk 3 rows 401 - 600	Chunk meta 200 rows a - ["otter","zebra"]

Attributes:

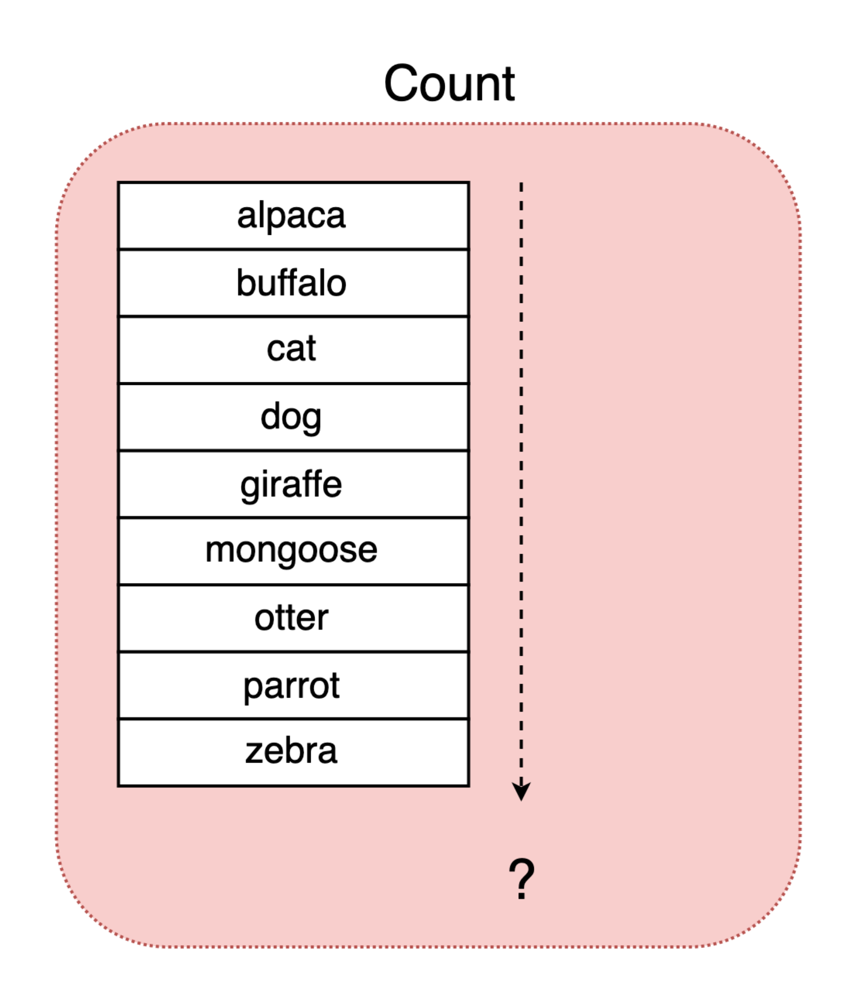
schema
a - String
b - Boolean
c - Integer

rowCount

...

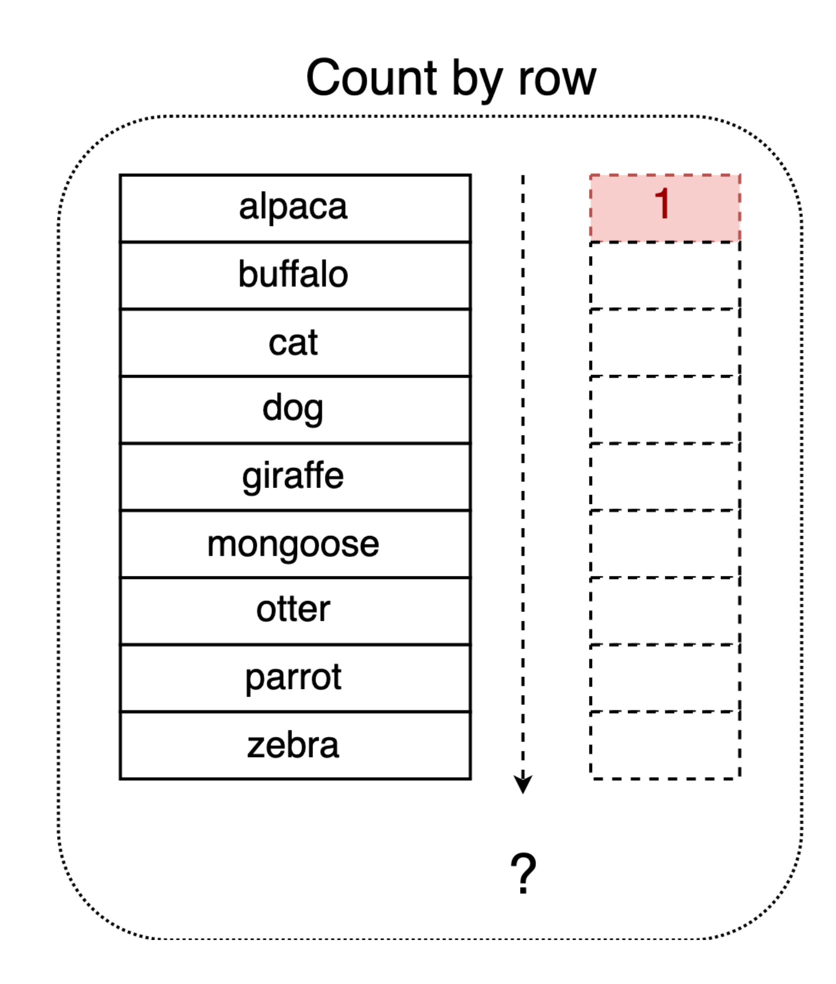
HL HighLoad++ Весна 2021

Батчи в Spark



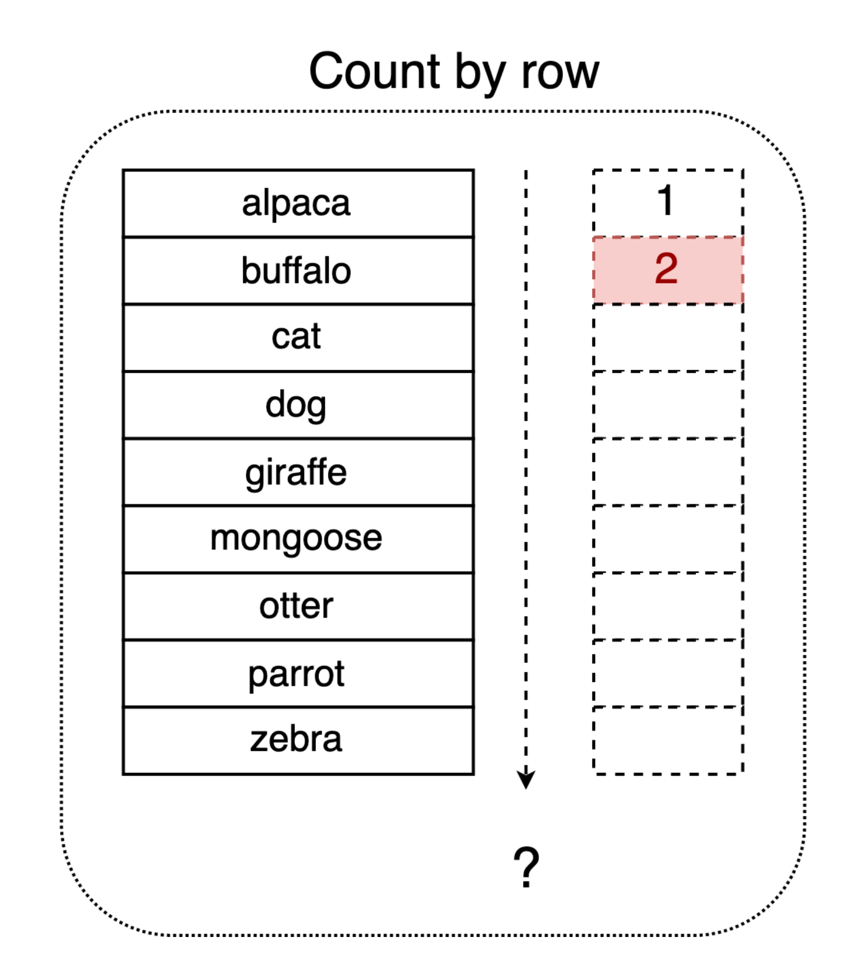
HighLoad++ Весна 2021

Построчный count



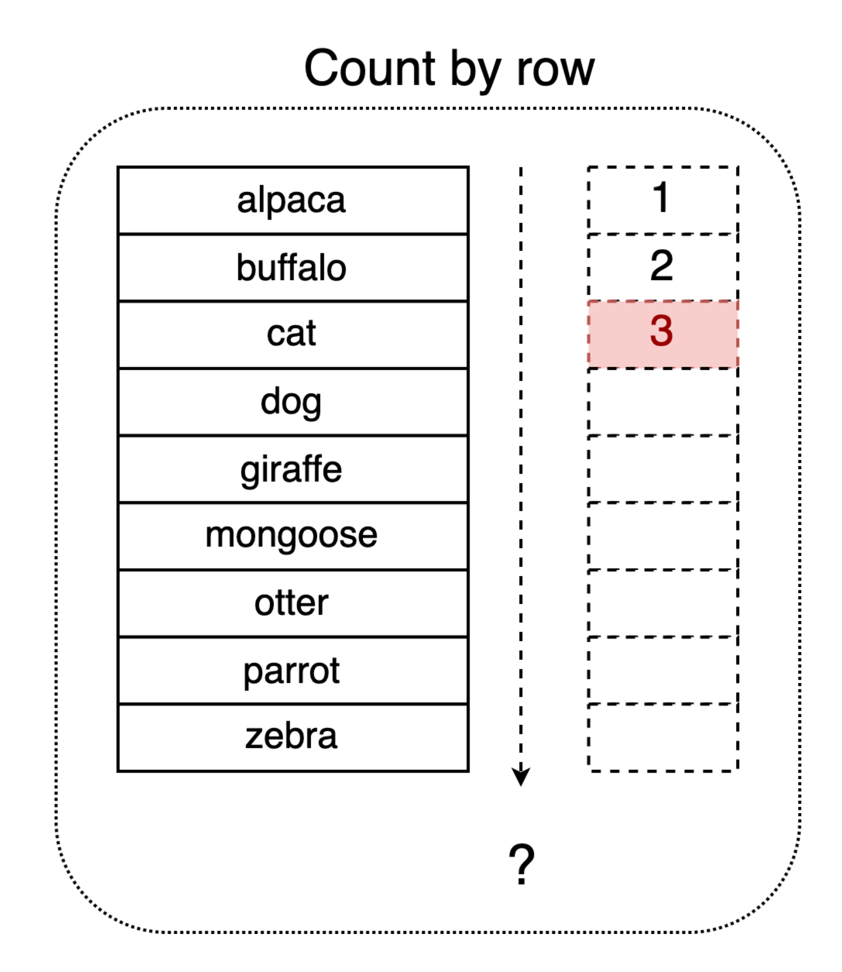
HL HighLoad++ Весна 2021

Построчный count



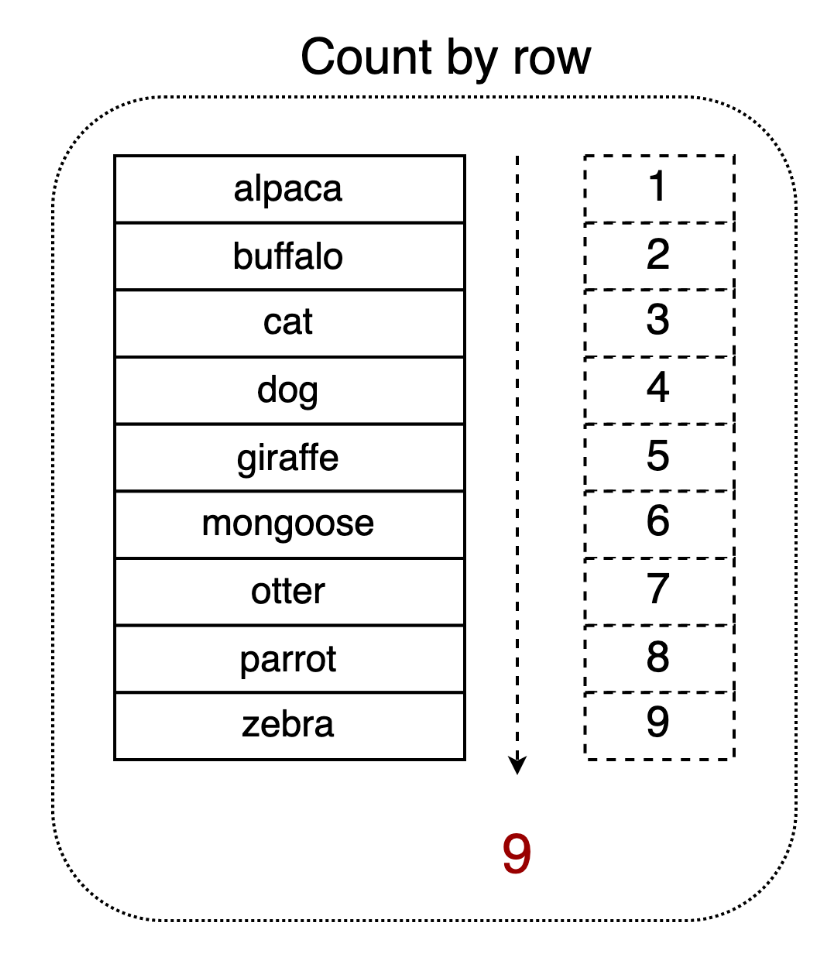
HL HighLoad++ Весна 2021

Построчный count



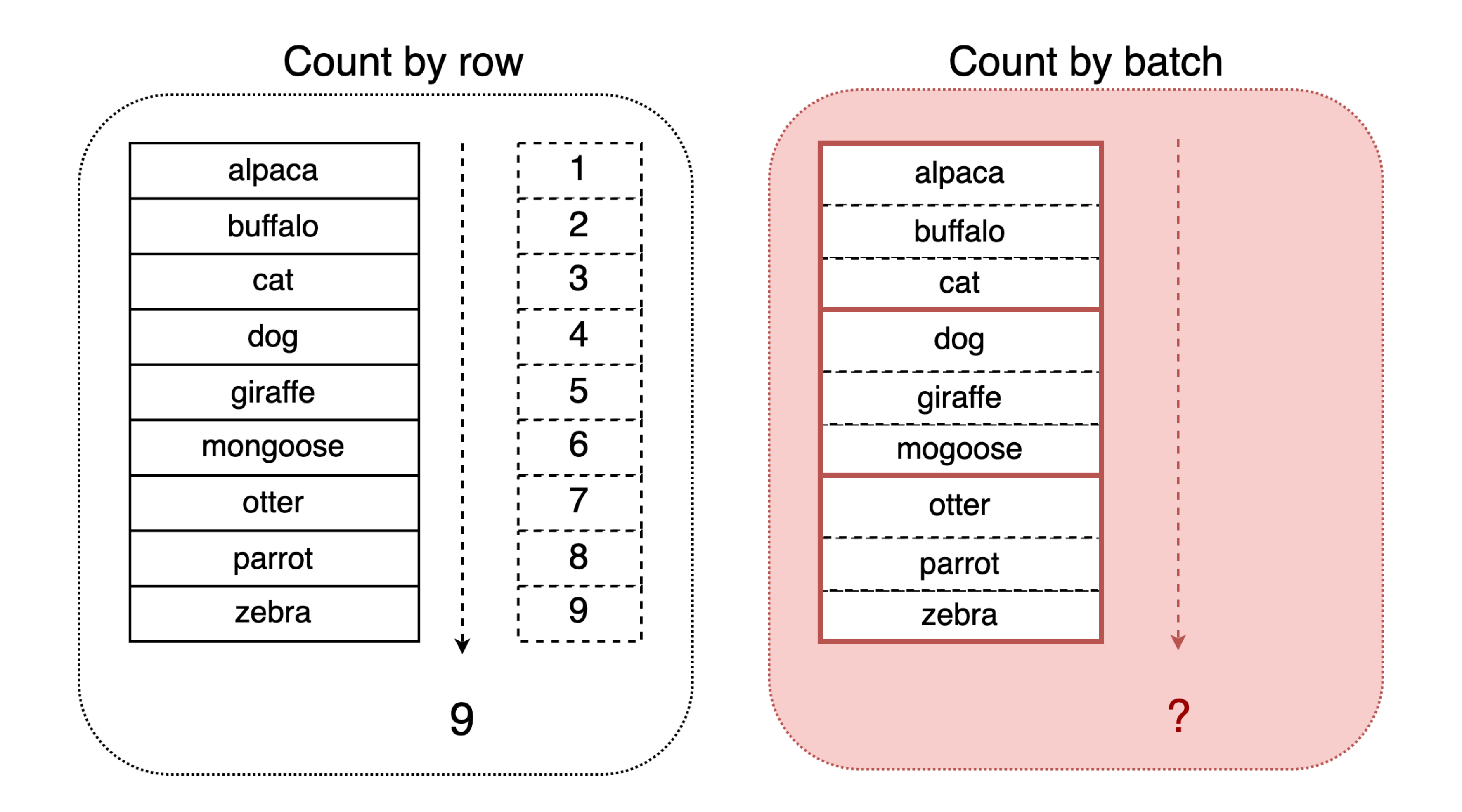
HL HighLoad++ Весна 2021

Построчный count

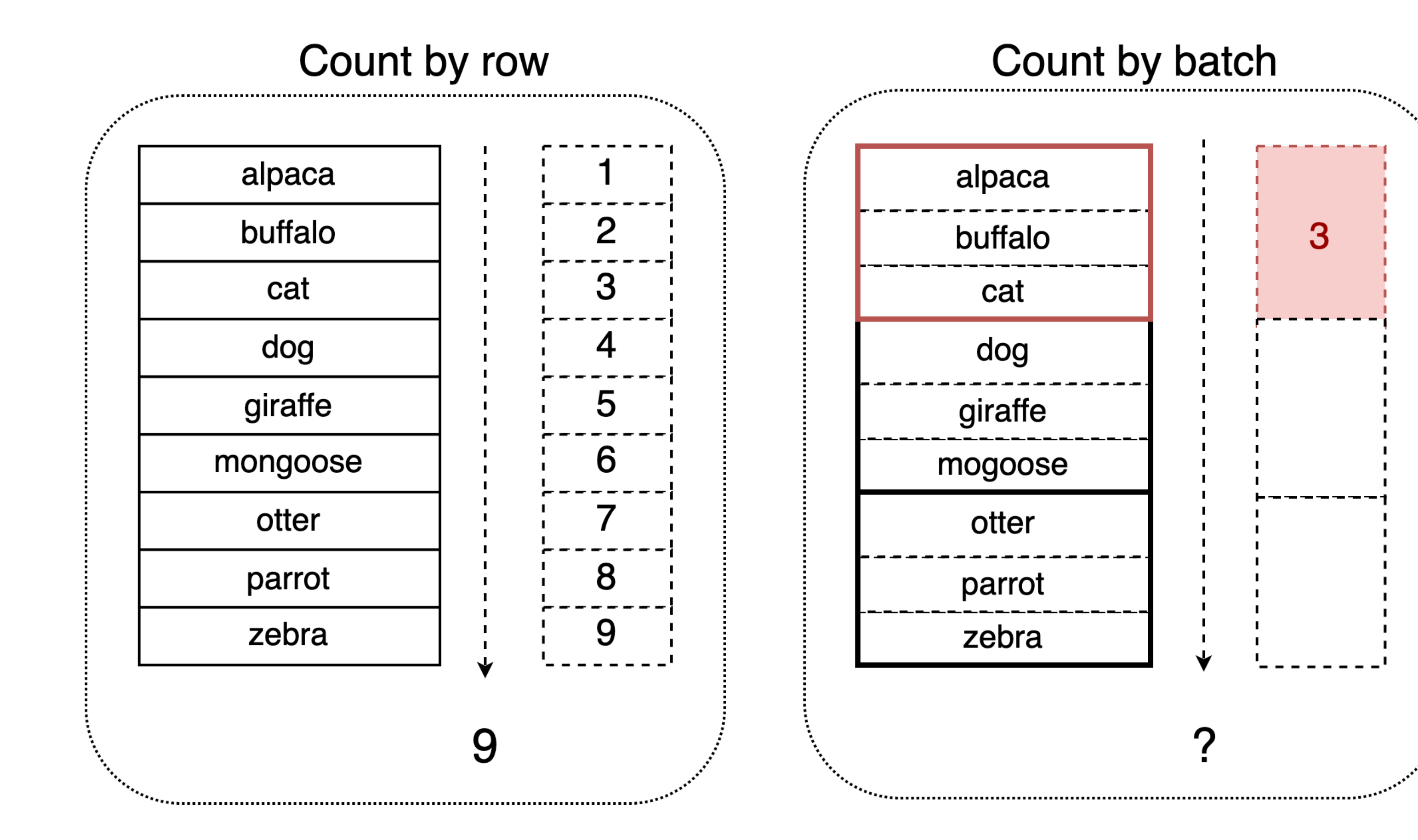


HL HighLoad++ Весна 2021

Побатчевый count

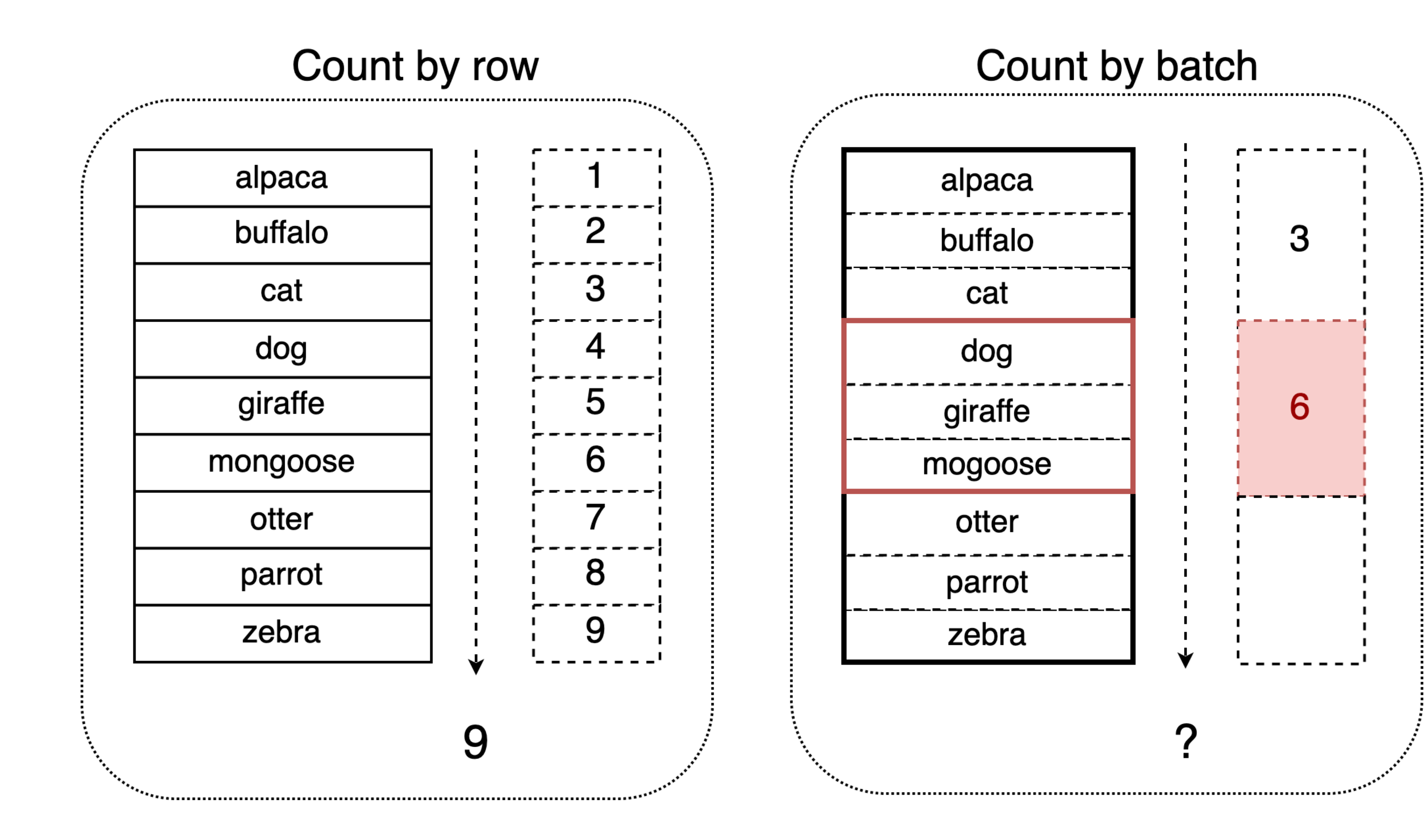


Побатчевый count



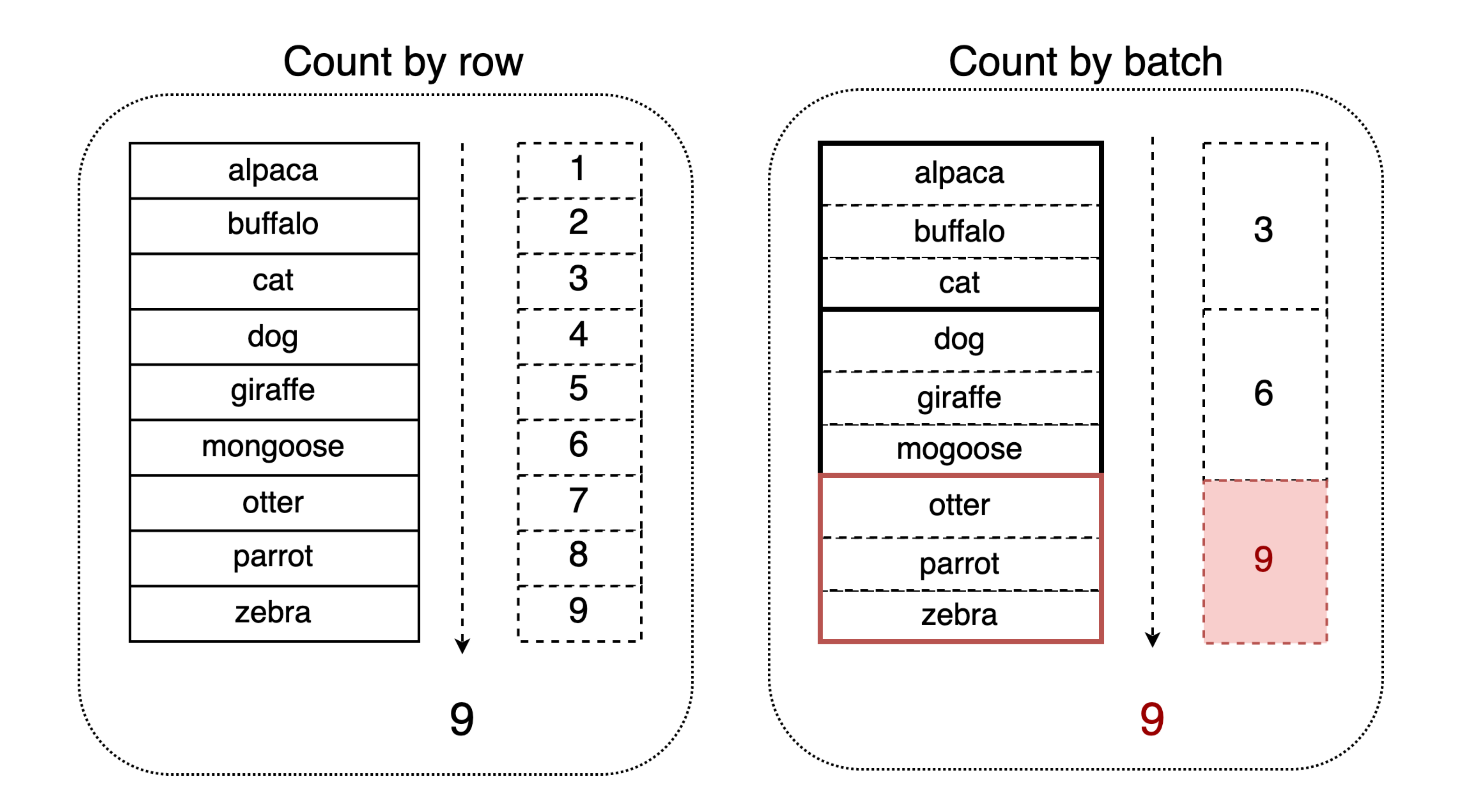
HL HighLoad++ Весна 2021

Побатчевый count



HL HighLoad++ Весна 2021

Побатчевый count



HL HighLoad++ Весна 2021

Почему плохо читать построчно

- Не даём спарку использовать метаданные батчей
- Тратим CPU на перекладку из поколоночного формата в построчный

HL HighLoad++ Весна 2021

Что не так с простым подходом

- Нет чтения батчами
- Нельзя прочитать директорию целиком

YT – файловая система

HDFS

```
/
|---> tmp
      |---> file.txt
      |---> example
            |---> dataset_1
                  |-----> part-0.parquet
                  |-----> part-1.parquet
                  |-----> part-2.parquet
            |---> dataset_2
                  |-----> part-0.parquet
                  |-----> part-1.parquet
```

YT

```
/
|---> tmp
      |---> file.txt
      |---> example
            |----> example_table_1
            |----> example_table_2
```


HL HighLoad++ Весна 2021

Чтение директории из HDFS

HDFS

```
/
|---> tmp
      |---> file.txt
      |---> example
            |---> dataset_1
                  |-----> part-0.parquet
                  |-----> part-1.parquet
                  |-----> part-2.parquet
            |---> dataset_2
                  |-----> part-0.parquet
                  |-----> part-1.parquet
```

```
spark.read.parquet("/tmp/example")
```

YT

```
/
|---> tmp
      |---> file.txt
      |---> example
            |----> example_table_1
            |----> example_table_2
```


HL HighLoad++ Весна 2021

Чтение директории из YT

HDFS

```
/
|---> tmp
      |---> file.txt
      |---> example
            |---> dataset_1
                  |-----> part-0.parquet
                  |-----> part-1.parquet
                  |-----> part-2.parquet
            |---> dataset_2
                  |-----> part-0.parquet
                  |-----> part-1.parquet
```

```
spark.read.parquet("/tmp/example")
```

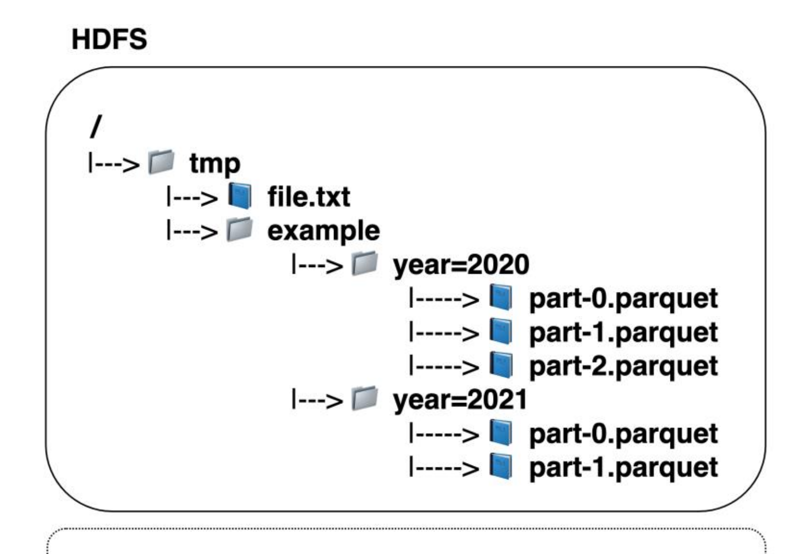
YT

```
/
|---> tmp
      |---> file.txt
      |---> example
            |----> example_table_1
            |----> example_table_2
```

```
spark.read.yt("/tmp/example")
```


HL HighLoad++ Весна 2021

Partition discovery

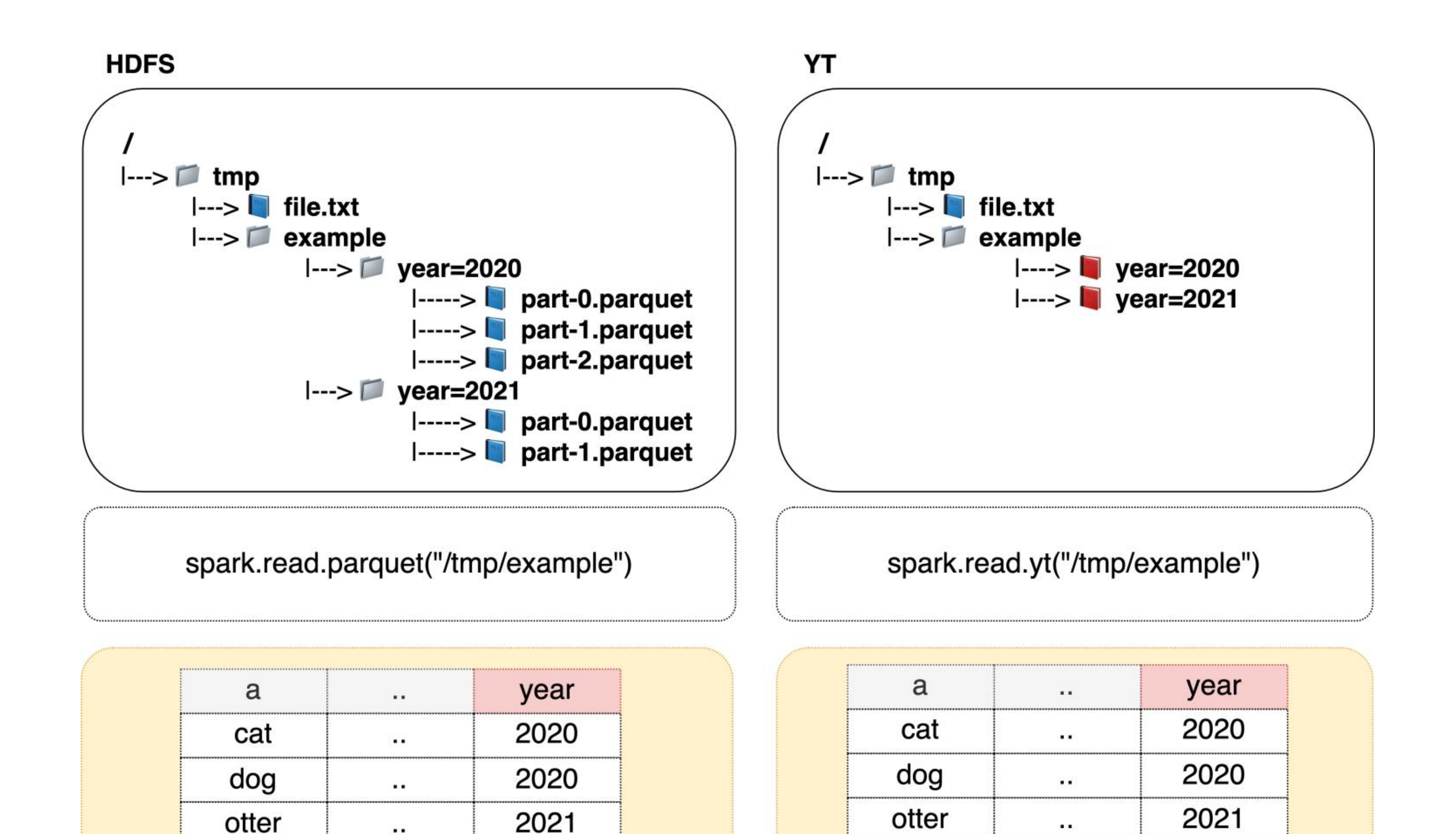



```
spark.read.parquet("/tmp/example")
```

a	..	year
cat	..	2020
dog	..	2020
otter	..	2021

HL HighLoad++ Весна 2021

Partition discovery



Чего мы хотим от нового подхода

- Научиться читать поколоночными батчами
- Переиспользовать код, написанный для HDFS и Parquet:
  - Partition discovery
  - Обход директорий на чтении
  - Создание / удаление временных файлов на записи

HL HighLoad++ Весна 2021

Новое решение

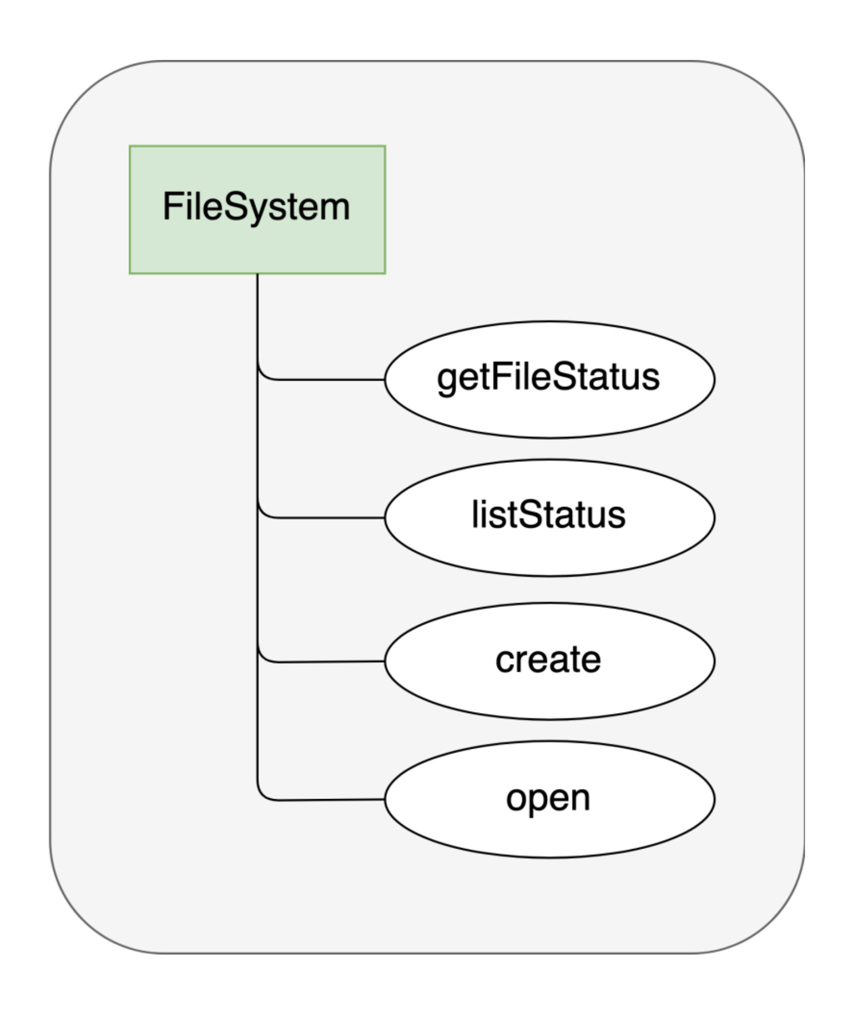
Как Spark читает Parquet с HDFS

Разделение DataSource на две части

- Данные лежат в файловой системе (например, HDFS)
- Данные сохранены в каком-то формате (например, Parquet)

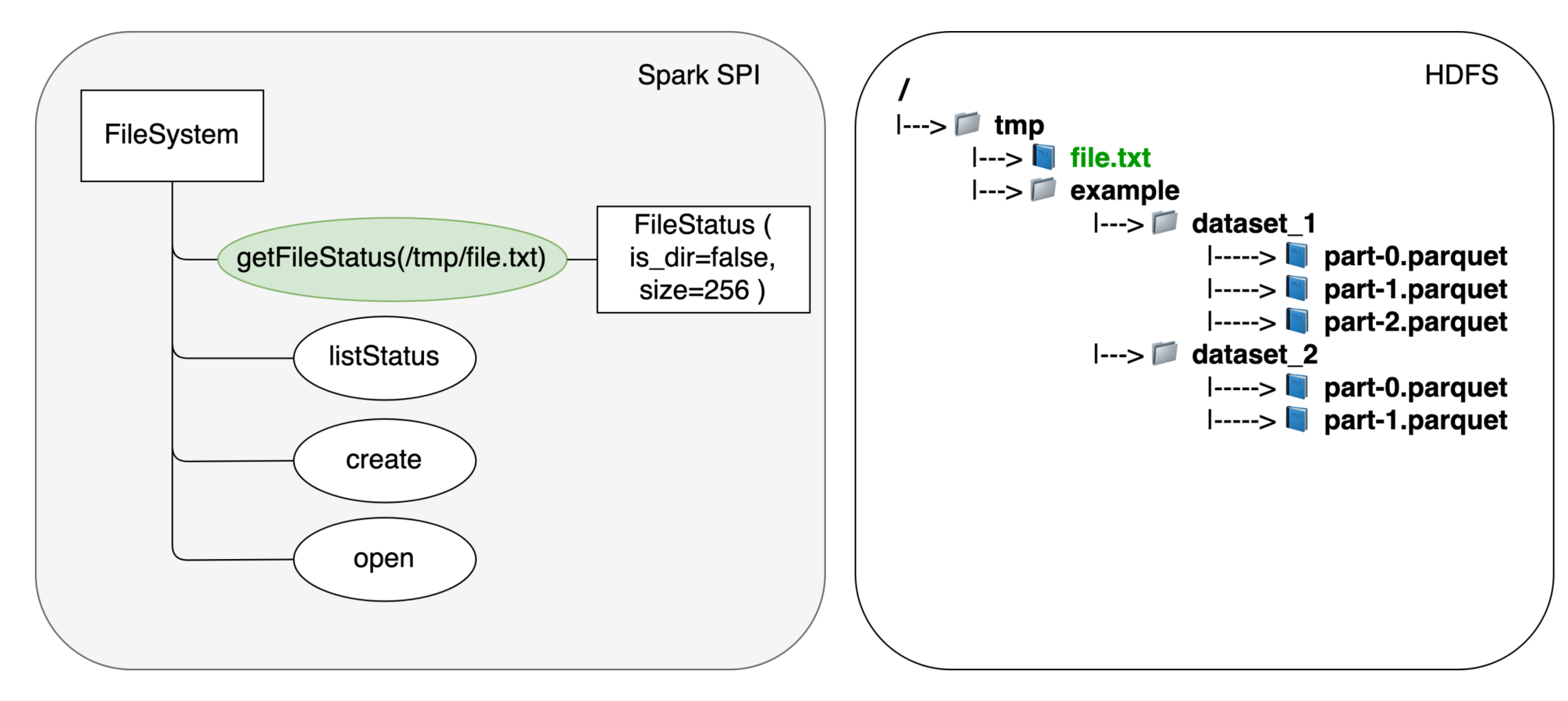
HL HighLoad++ Весна 2021

FileSystem



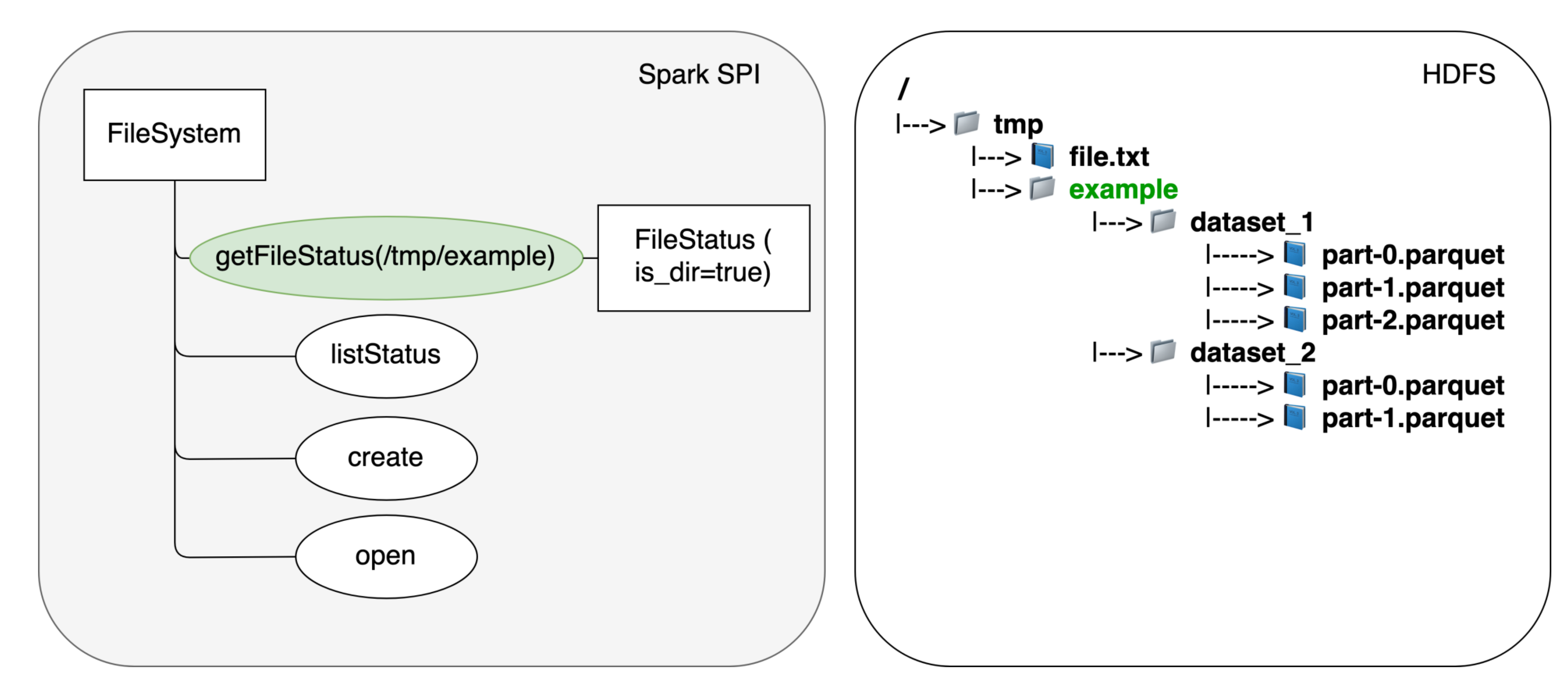
HL HighLoad++ Весна 2021

Статус файла



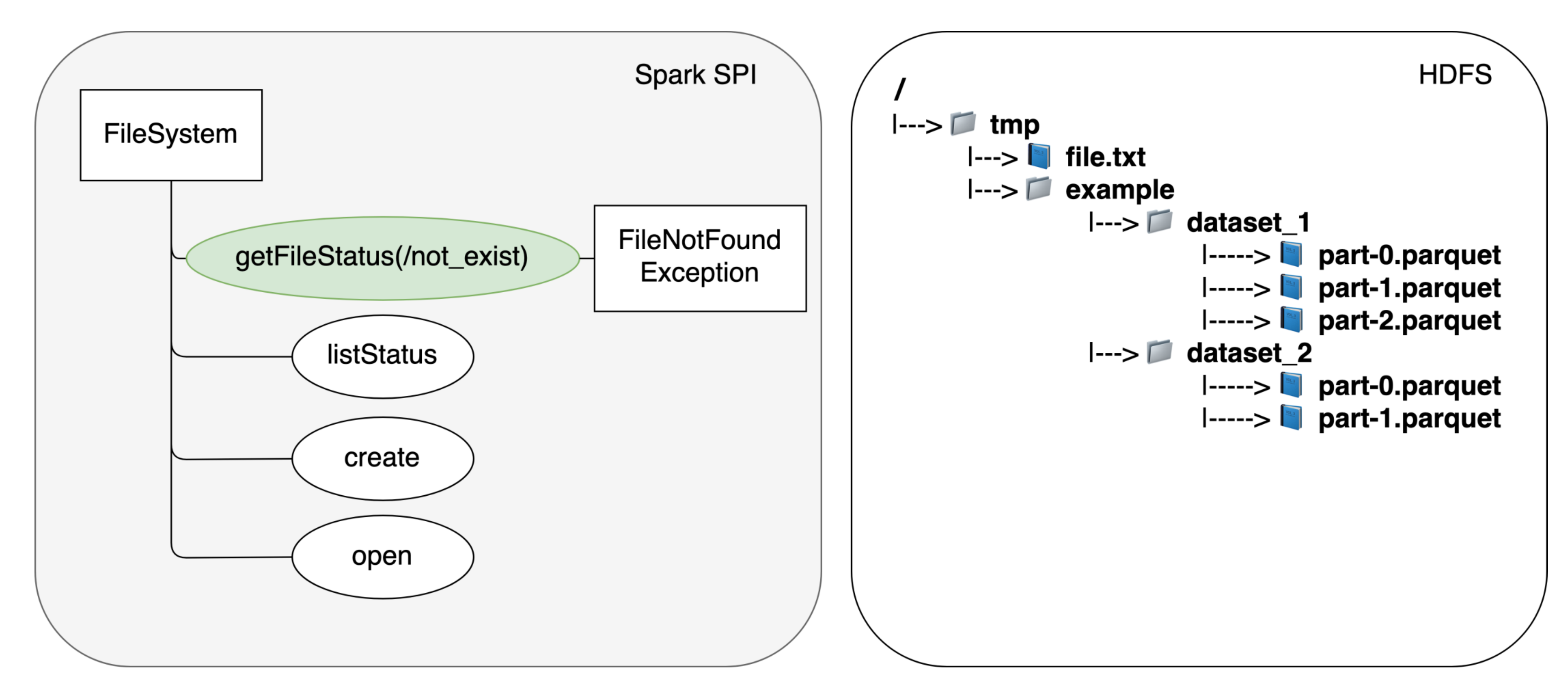
HL HighLoad++ Весна 2021

Статус директории



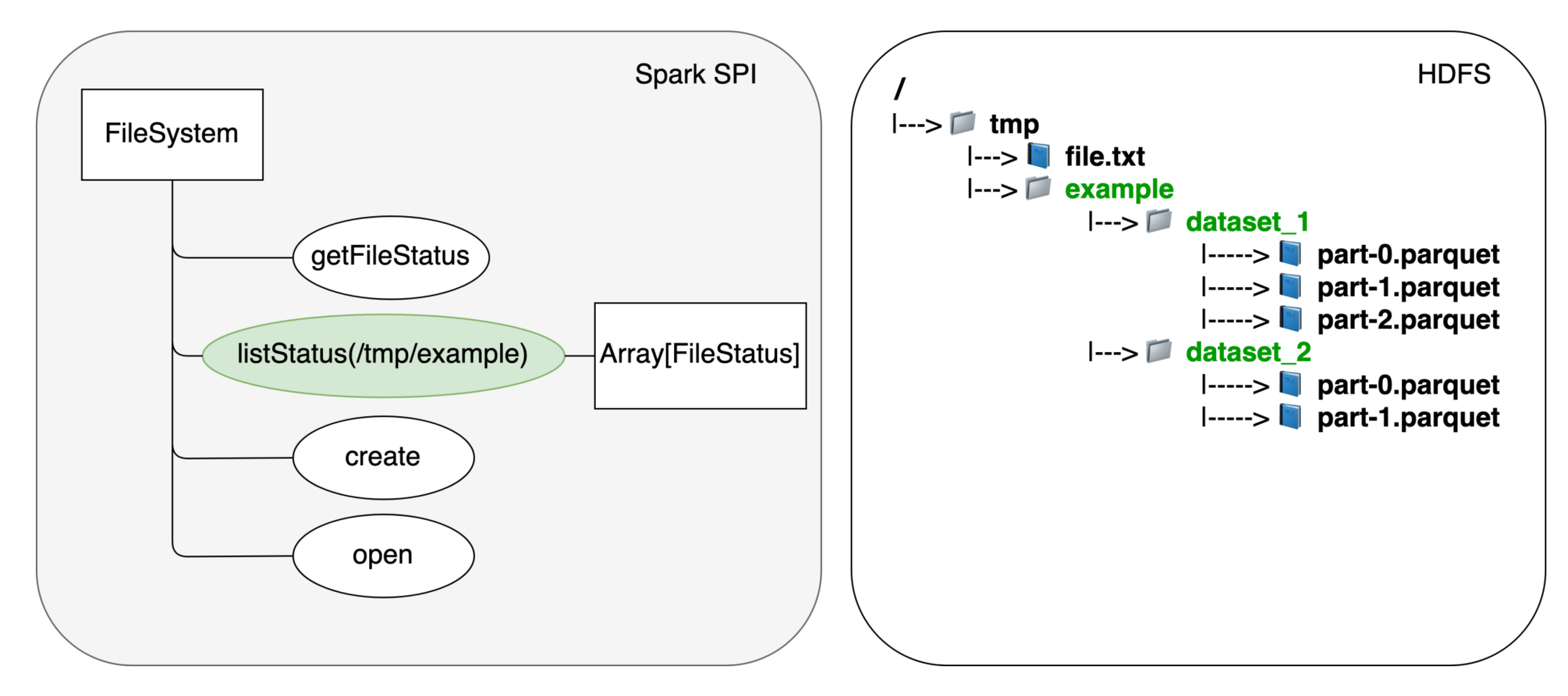
HL HighLoad++ Весна 2021

Файл не существует



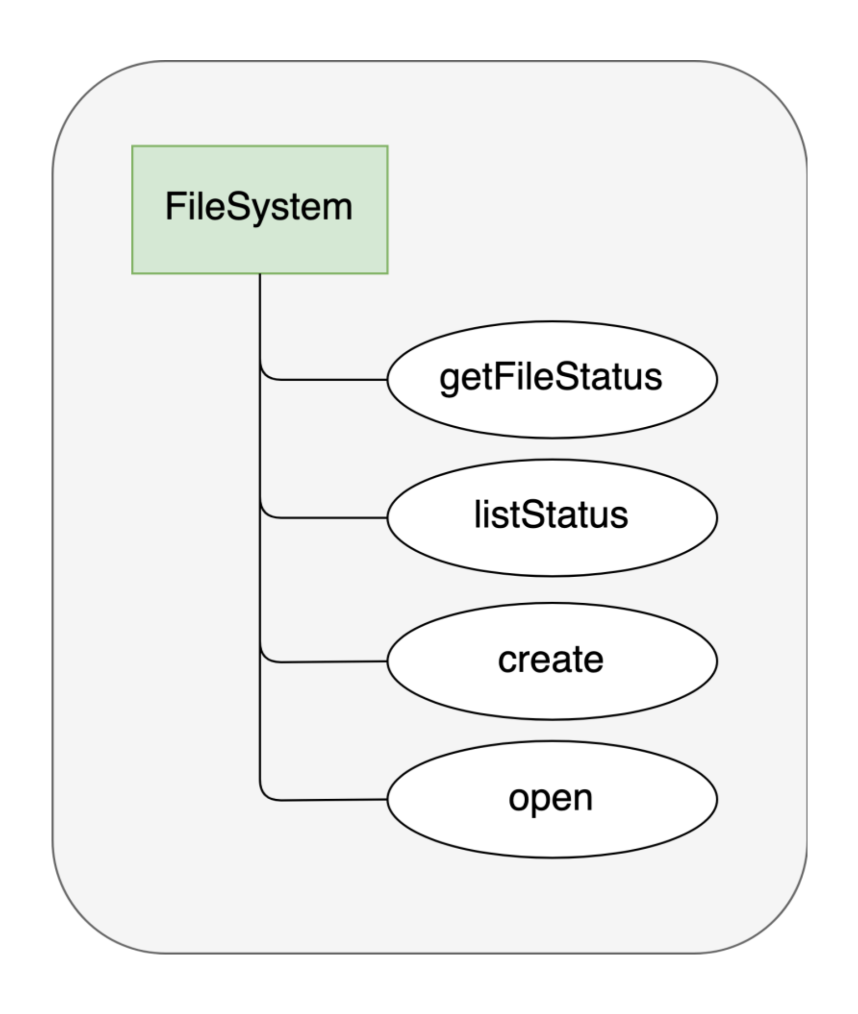
HL HighLoad++ Весна 2021

Листинг директории



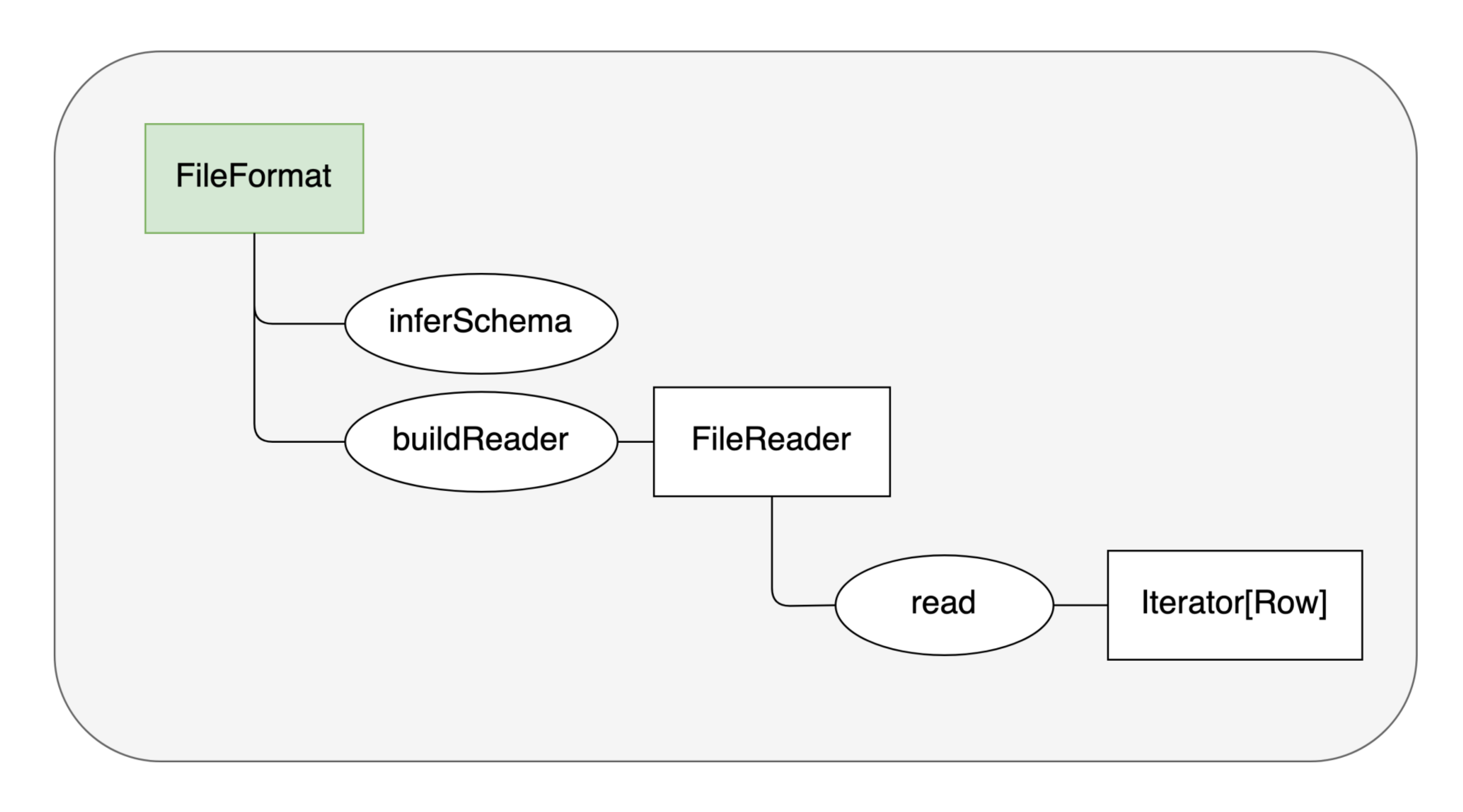
HL HighLoad++ Весна 2021

FileSystem



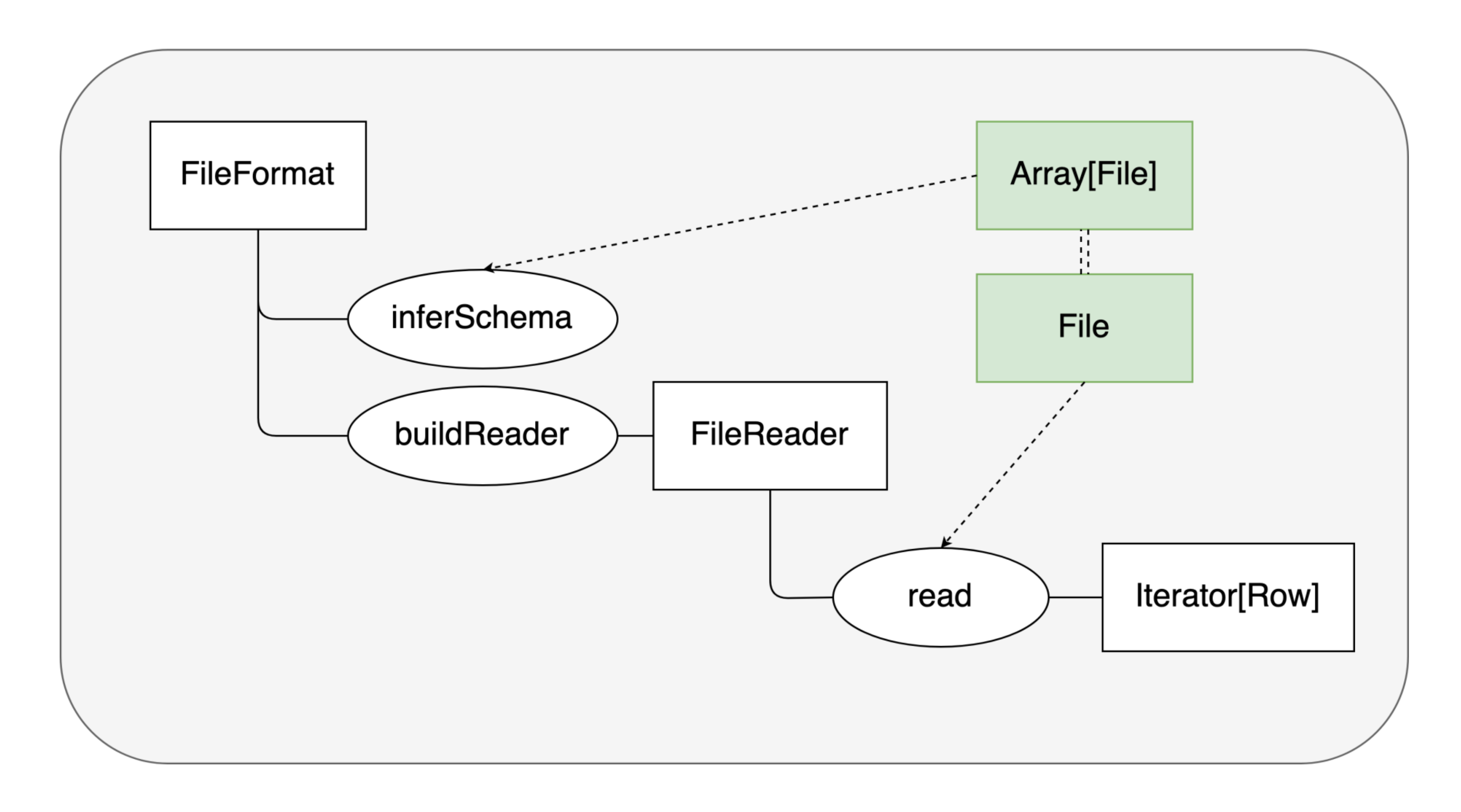
HL HighLoad++ Весна 2021

FileFormat



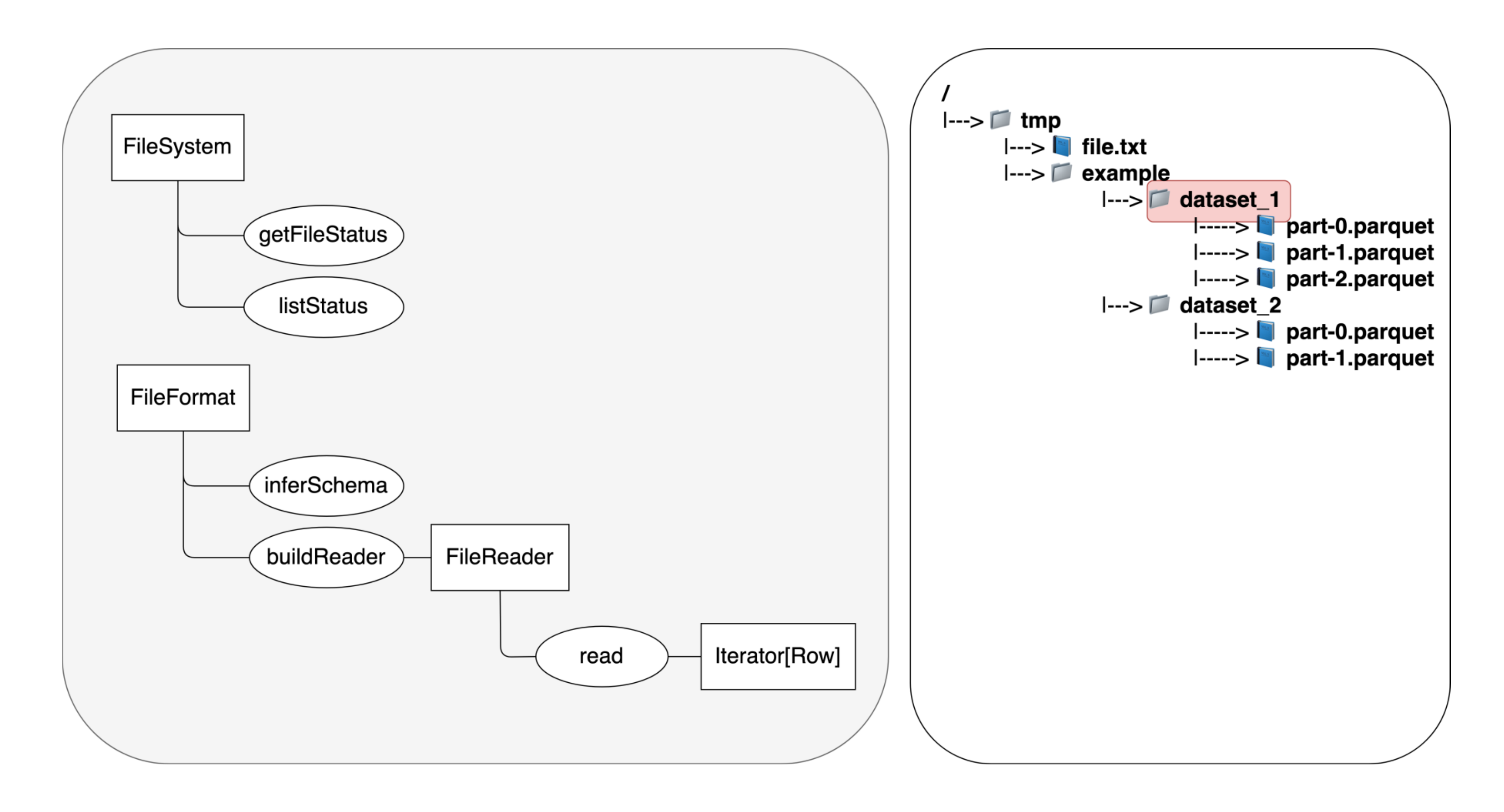
HL HighLoad++ Весна 2021

FileFormat



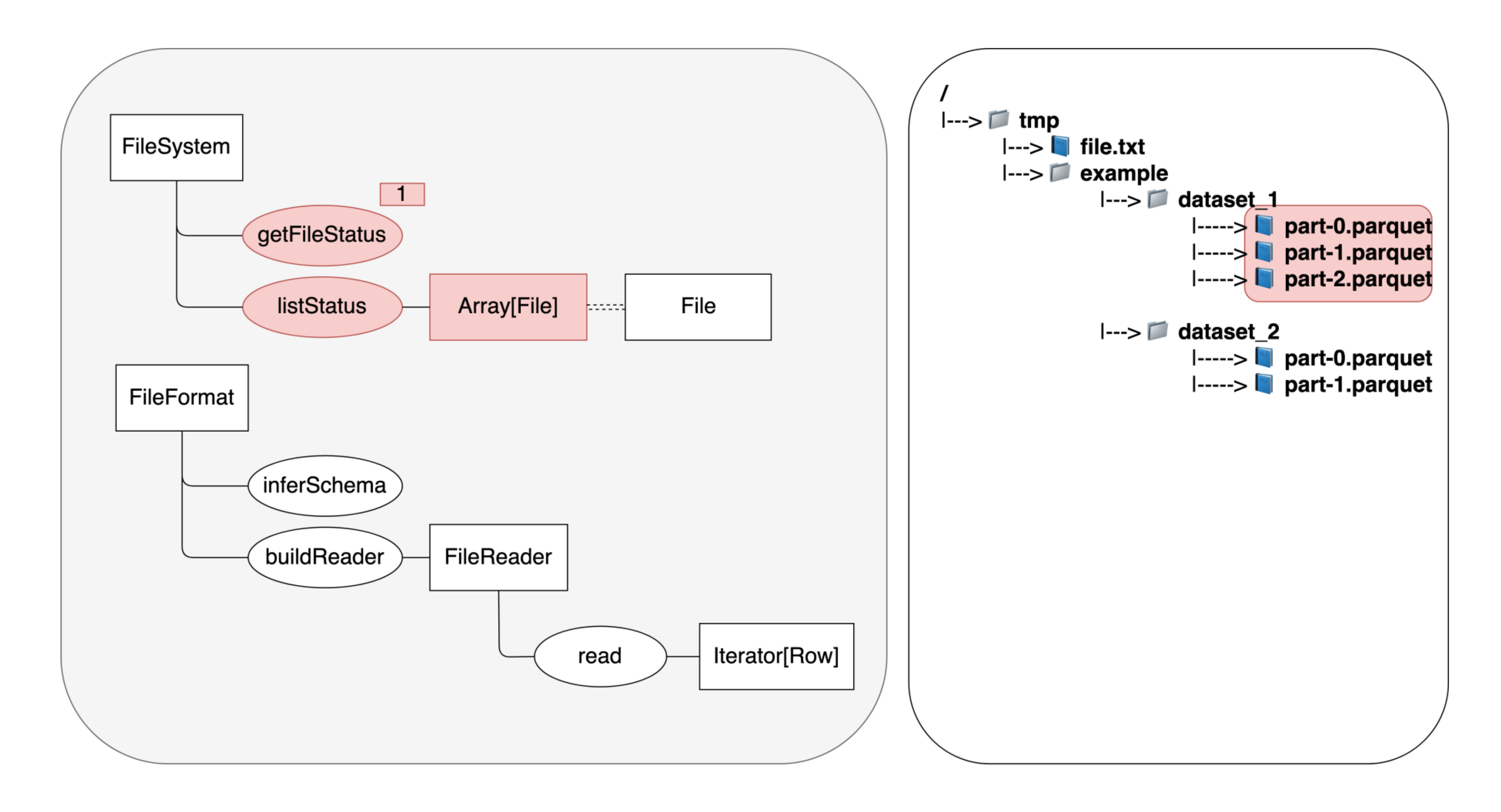
HL HighLoad++ Весна 2021

Чтение с HDFS



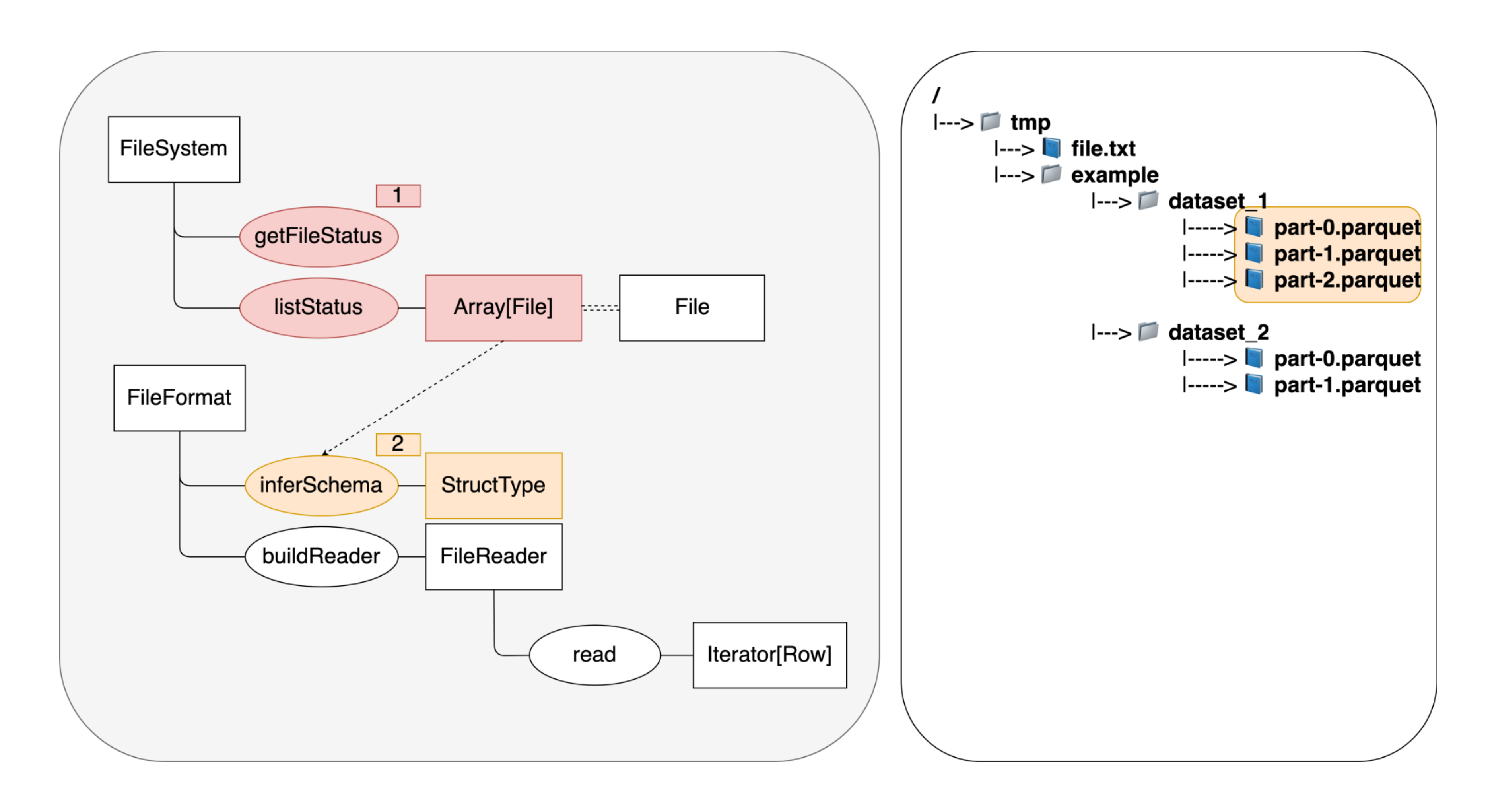
HL HighLoad++ Весна 2021

Листинг директории



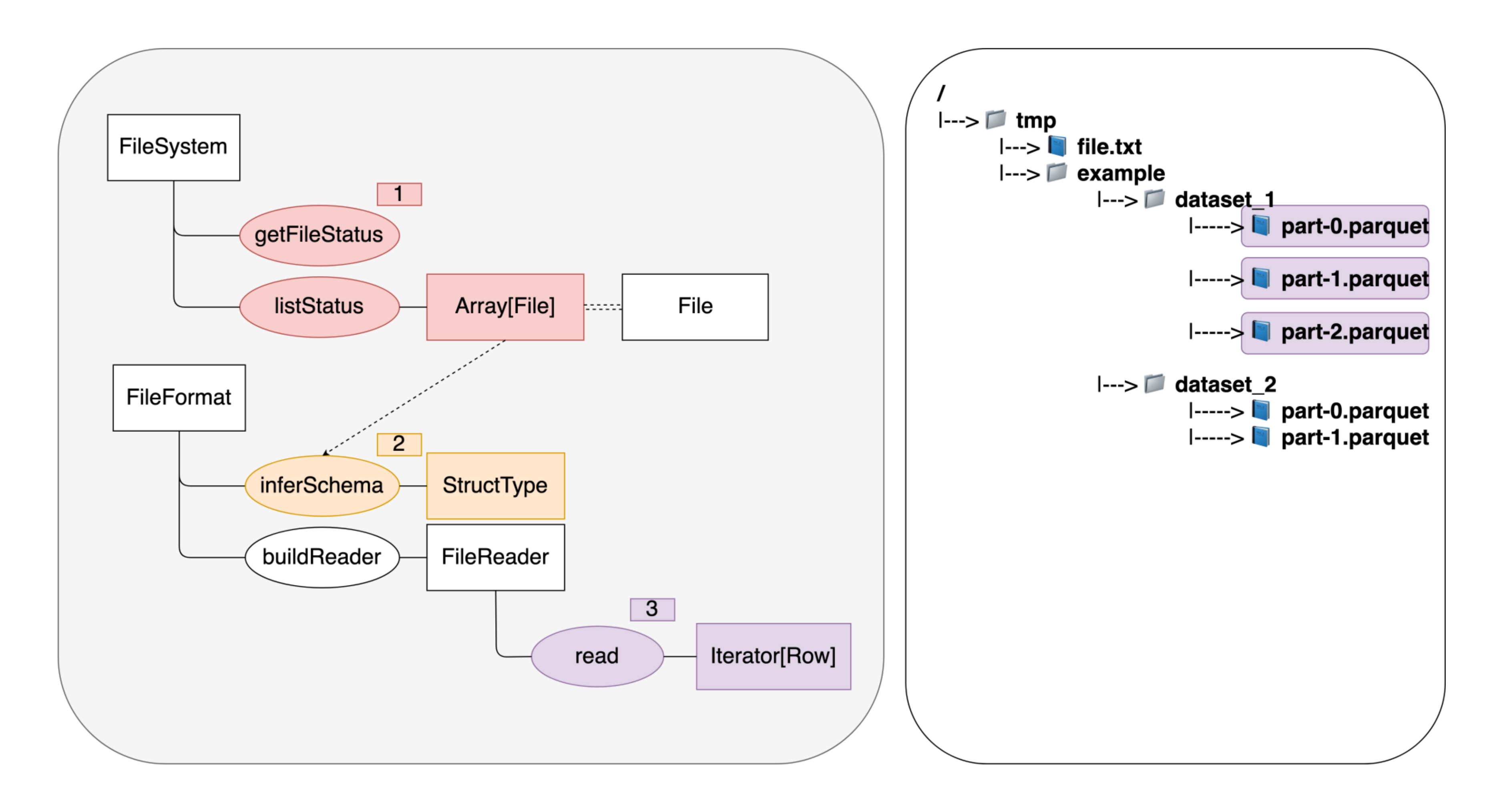
HL HighLoad++ Весна 2021

Чтение схемы



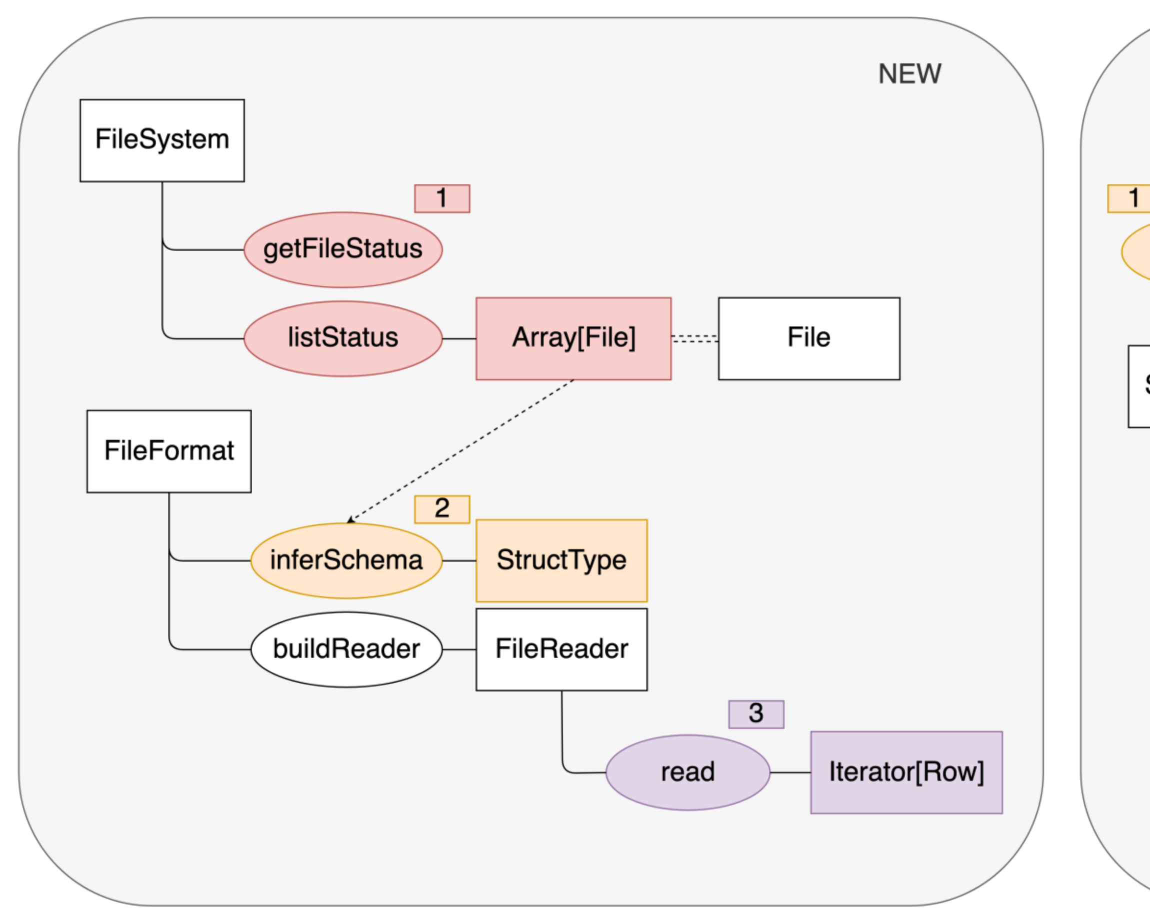
HL HighLoad++ Весна 2021

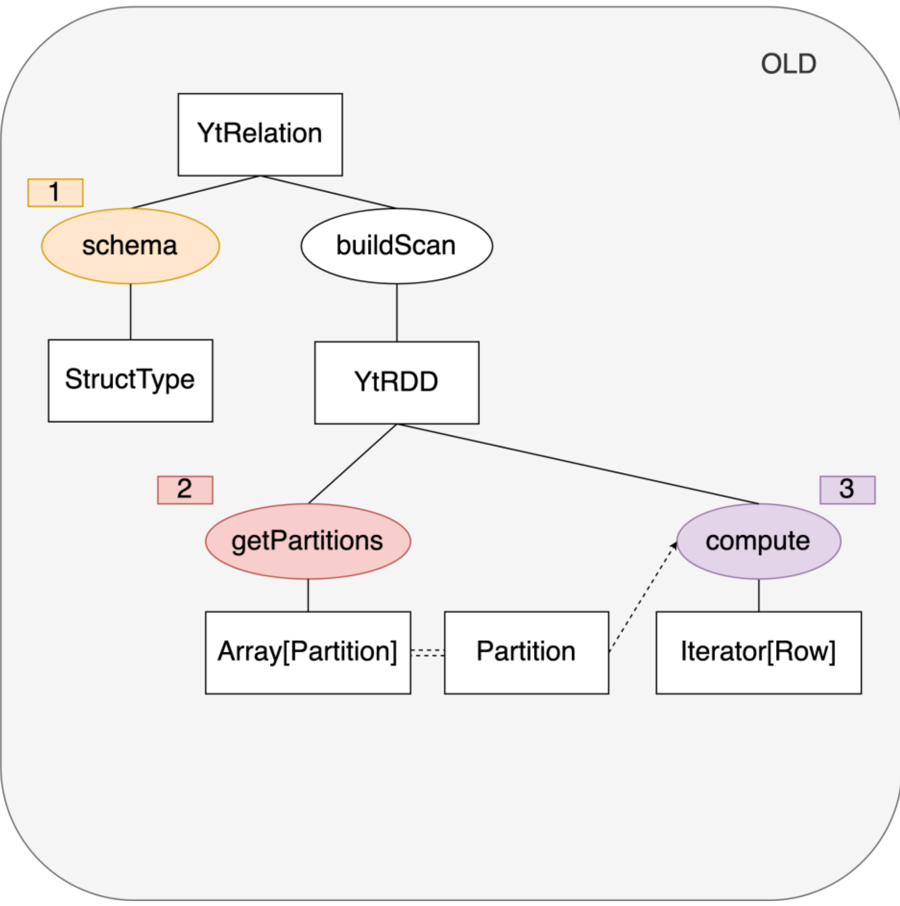
Чтение партиции



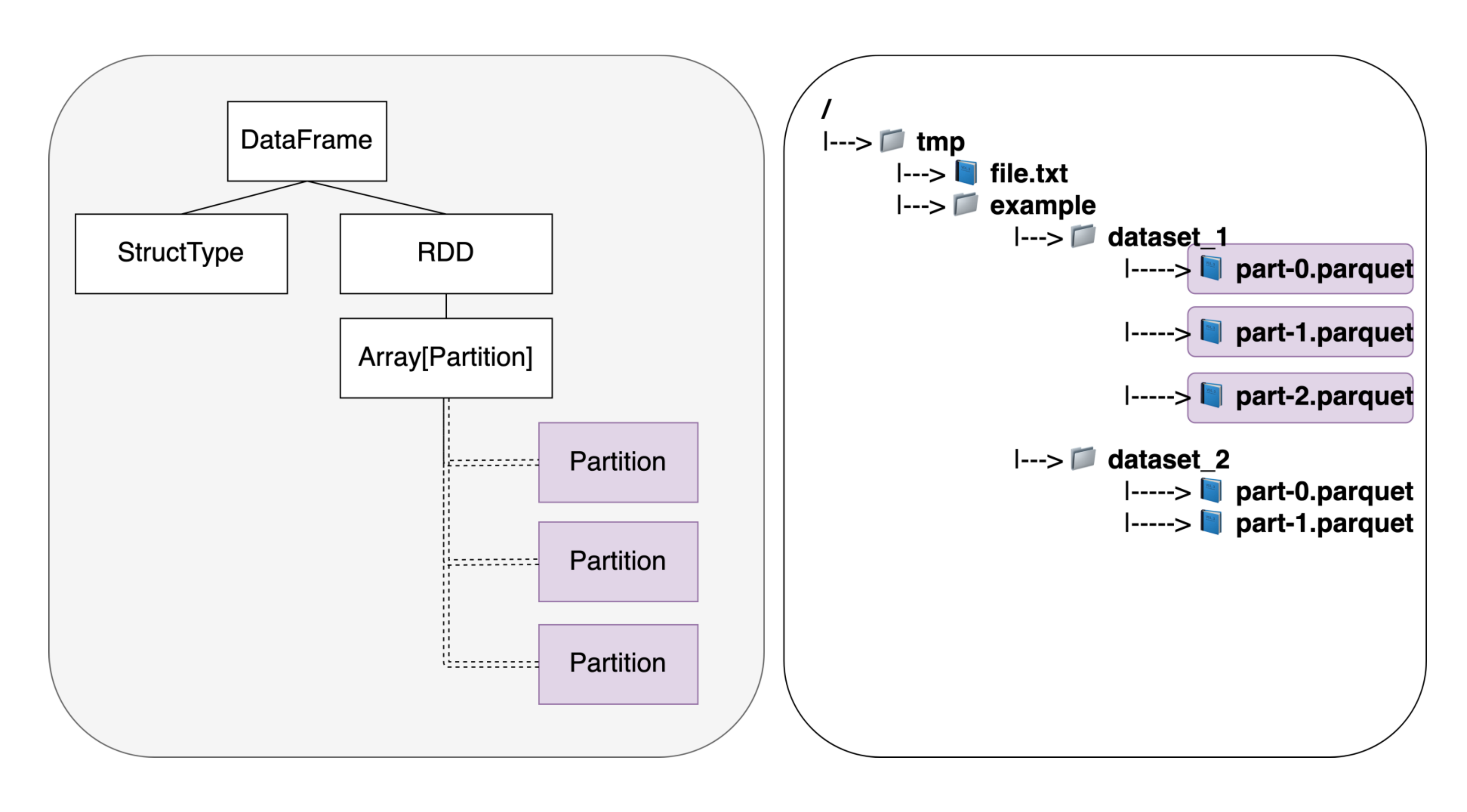
HL HighLoad++ Весна 2021

Сравнение с простым подходом



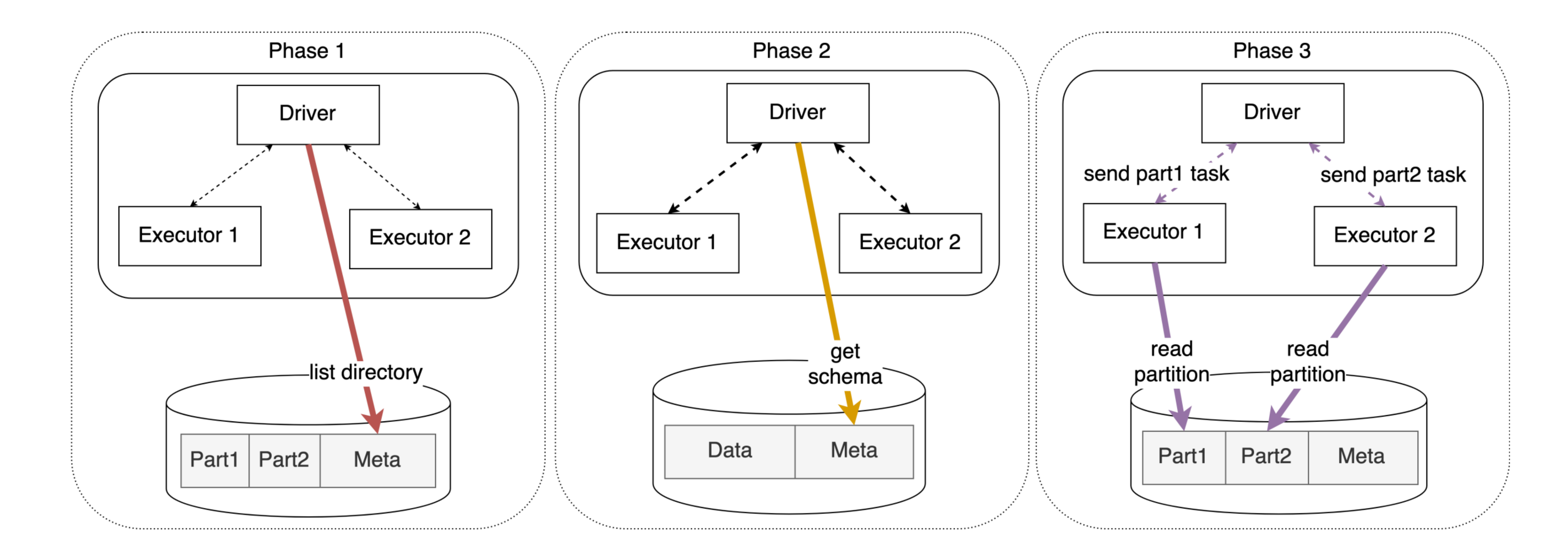


Файл = партиция



HL HighLoad++ Весна 2021

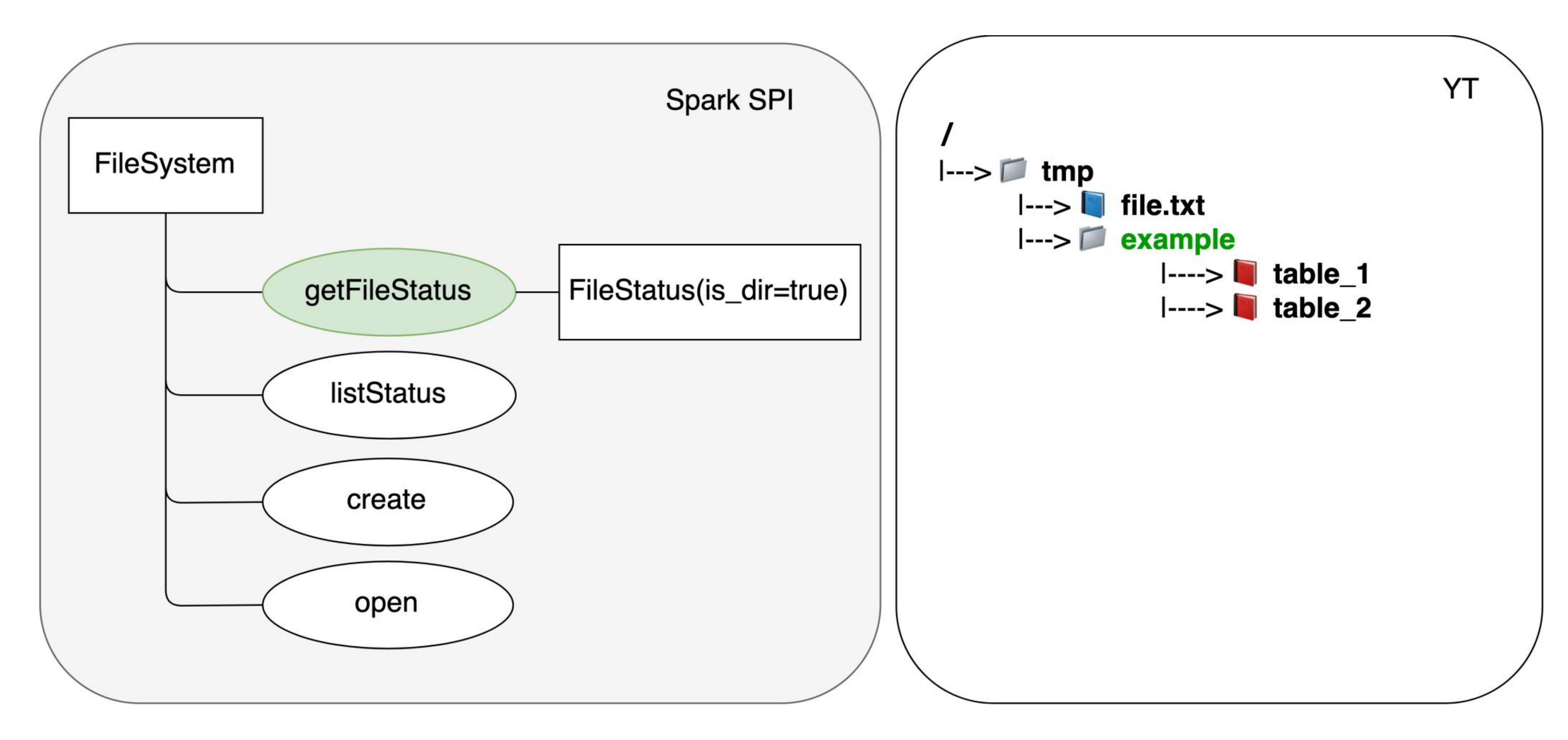
Как это выполняется



HL HighLoad++ Весна 2021

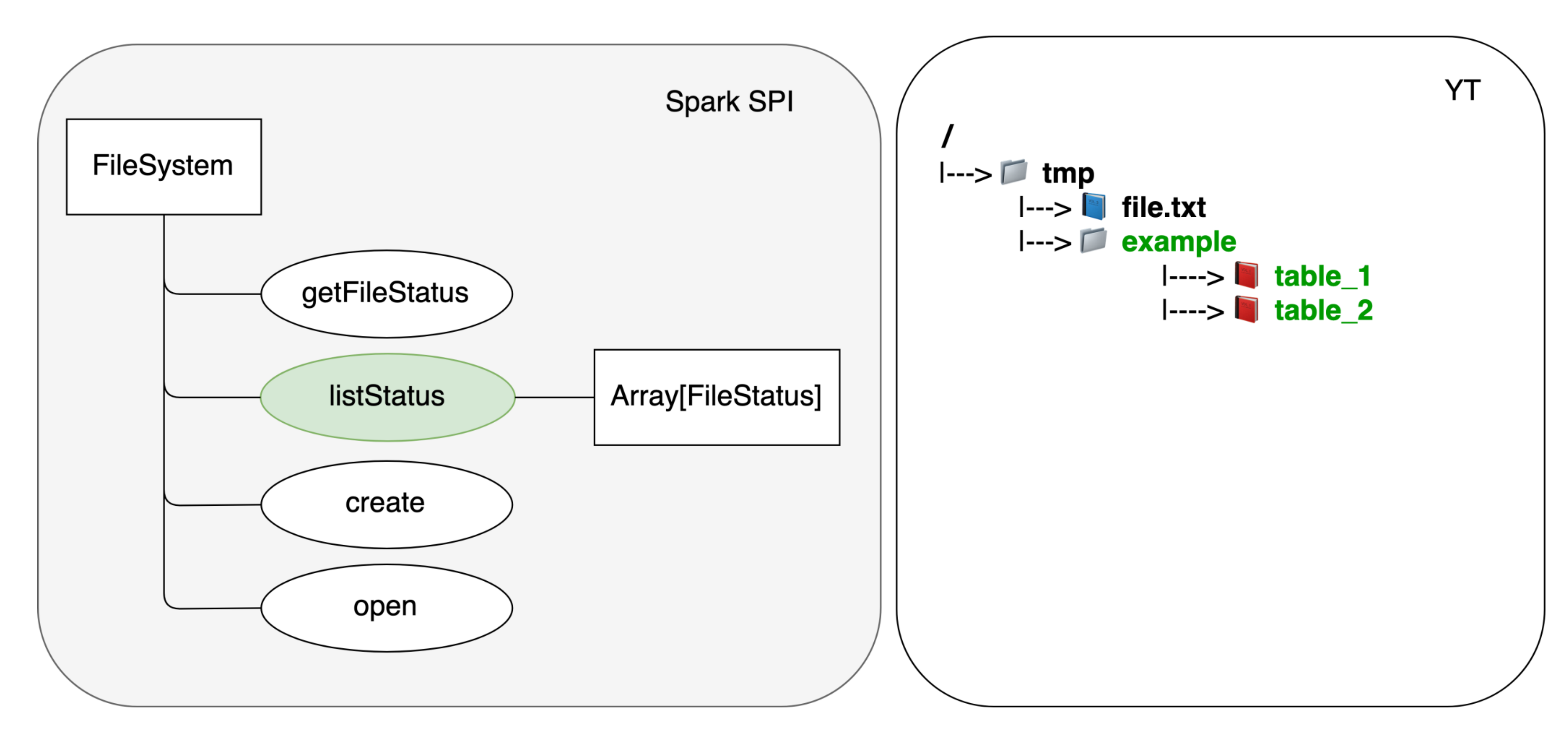
Как SPYT читает таблицы с YT

YtFileSystem



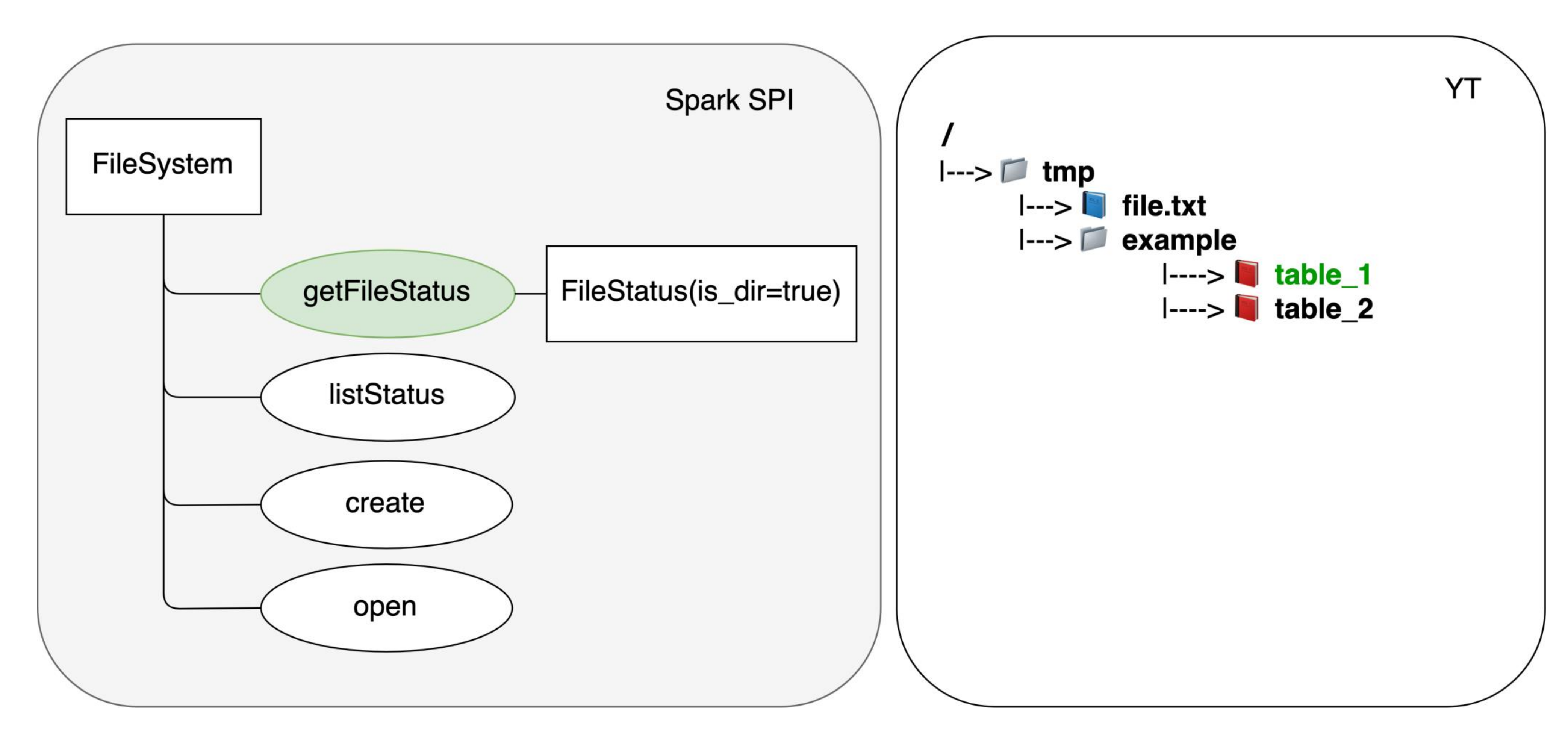
HL HighLoad++ Весна 2021

Листинг директории



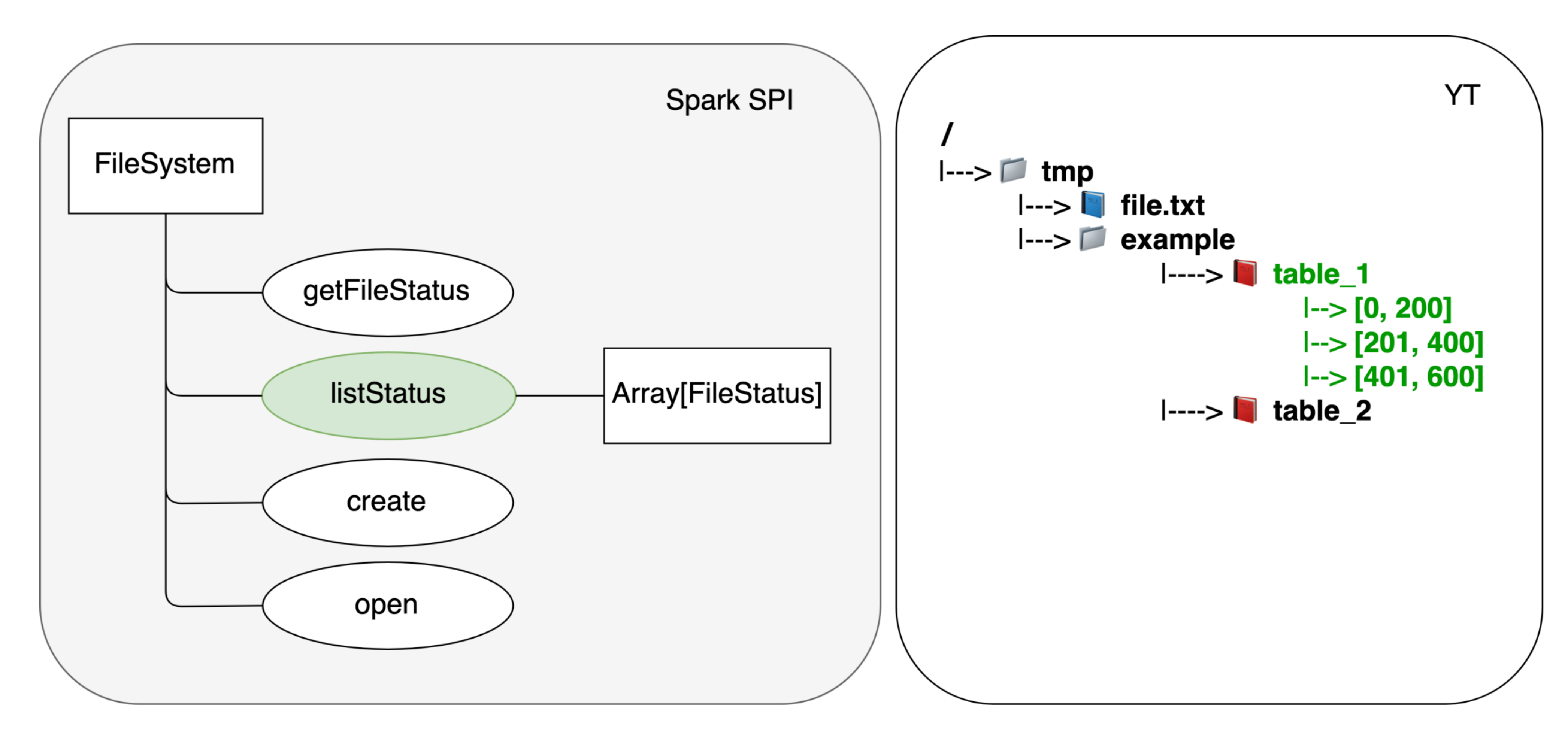
HL HighLoad++ Весна 2021

Таблица = директория



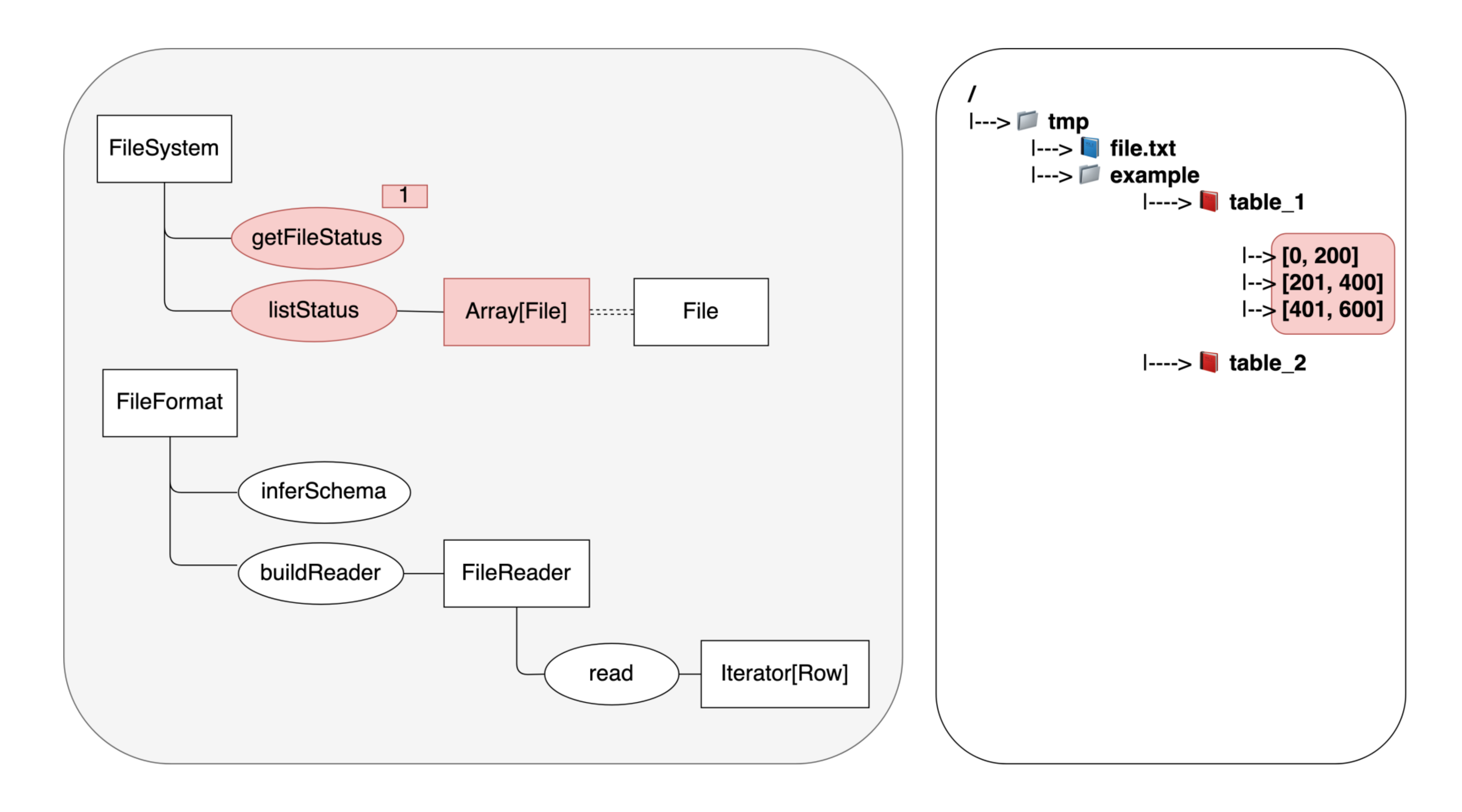
HL HighLoad++ Весна 2021

Псевдолистинг таблицы



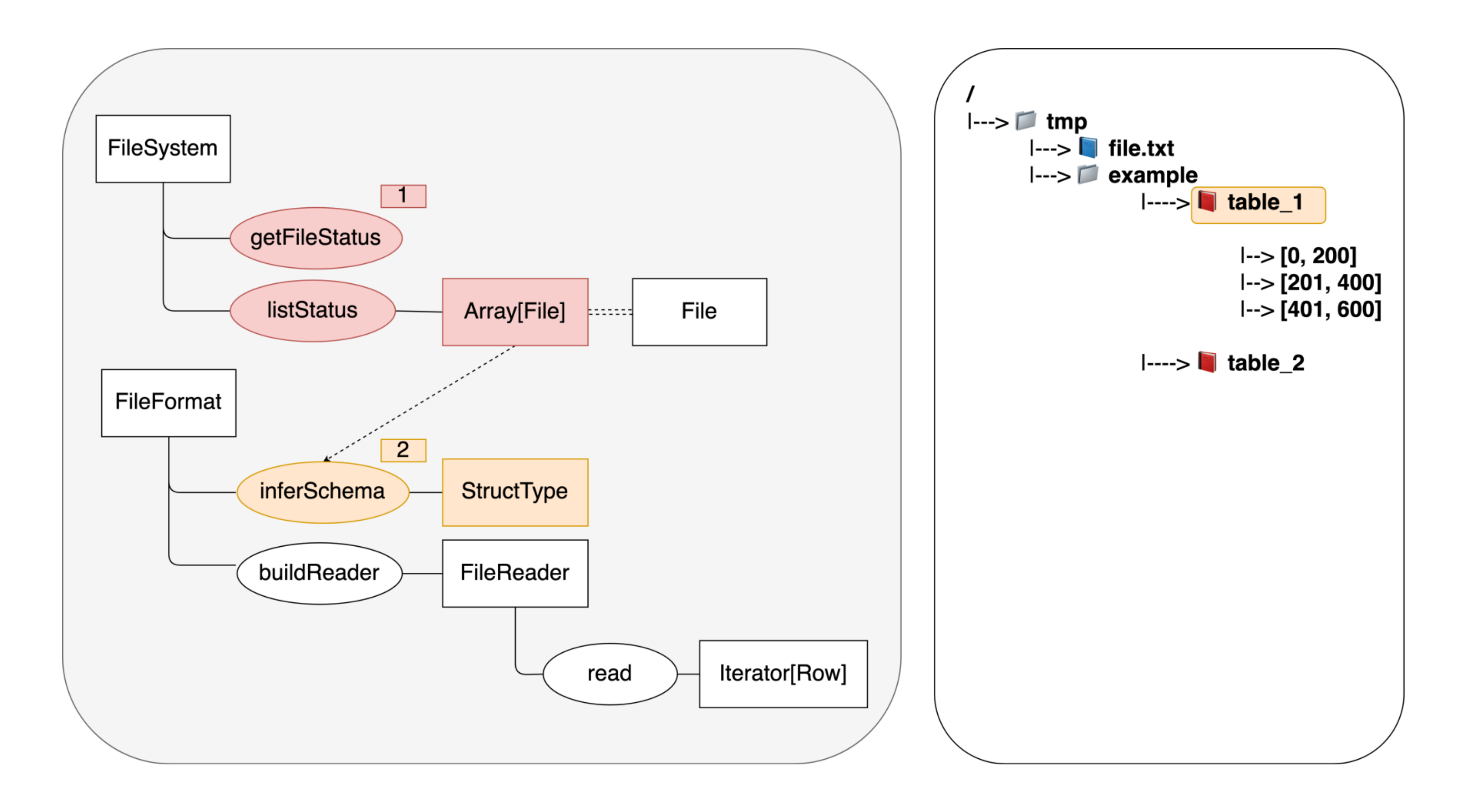
HL HighLoad++ Весна 2021

Разбиваем таблицу на партиции



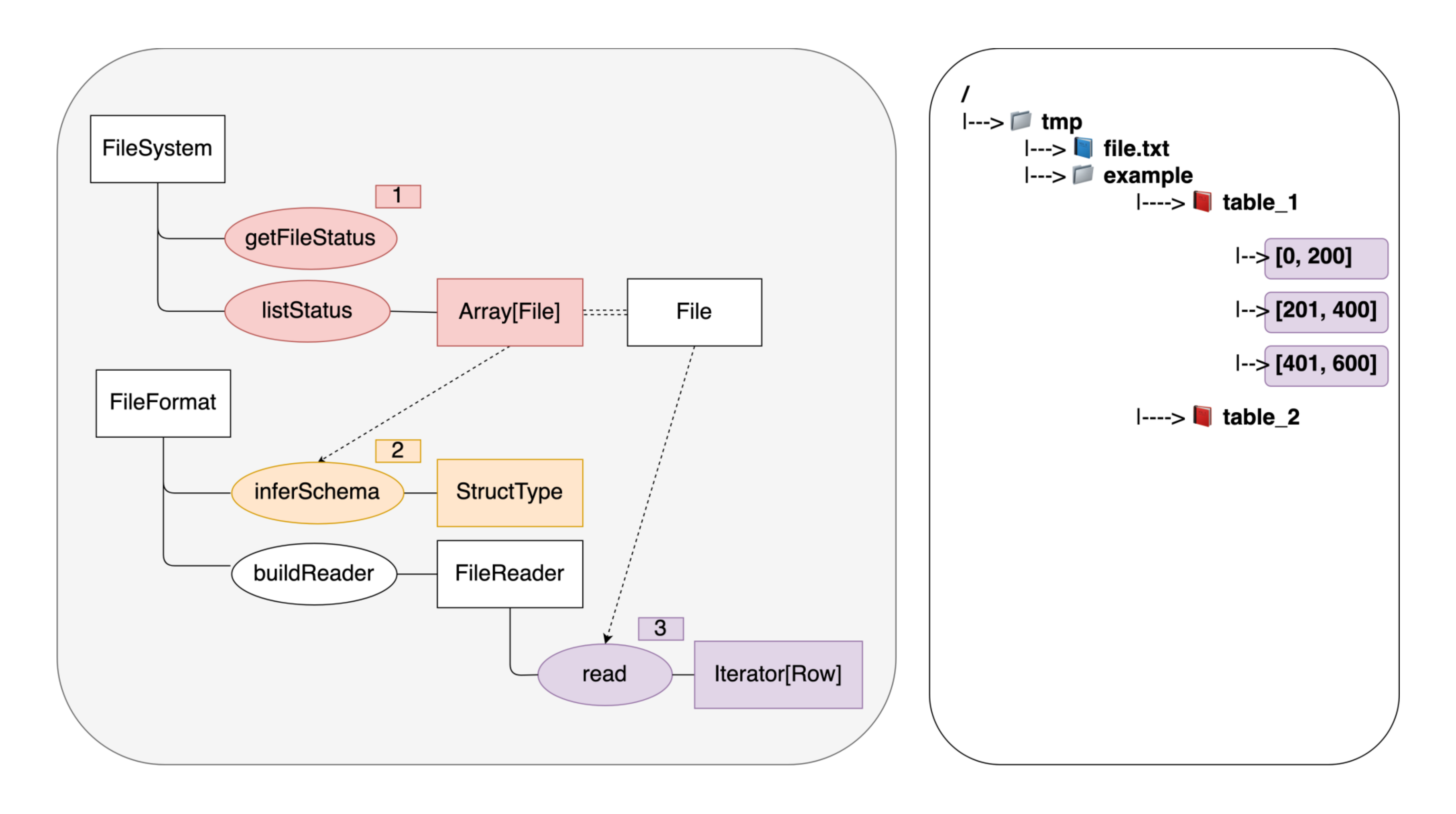
HL HighLoad++ Весна 2021

Читаем схему



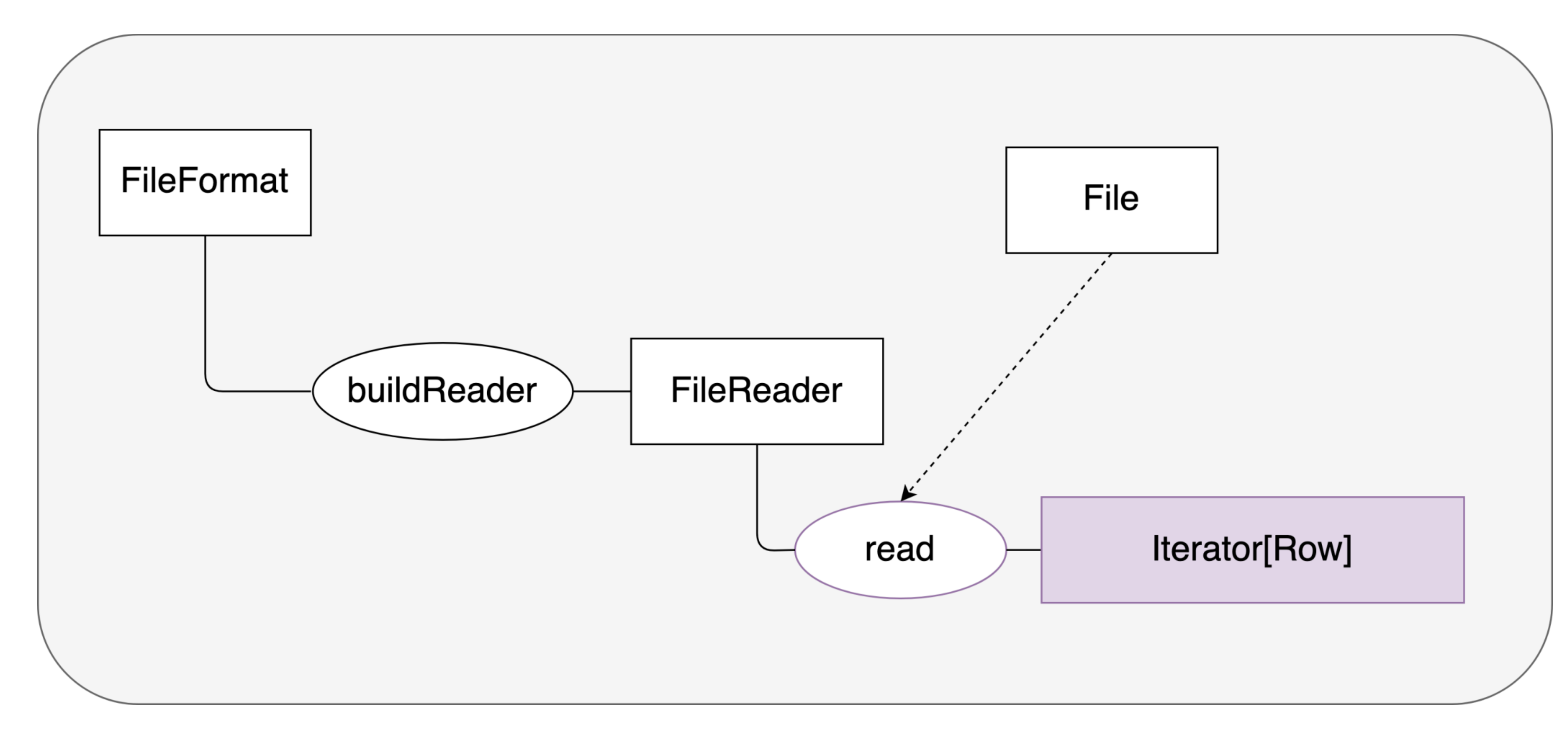
HL HighLoad++ Весна 2021

Читаем отдельные партиции



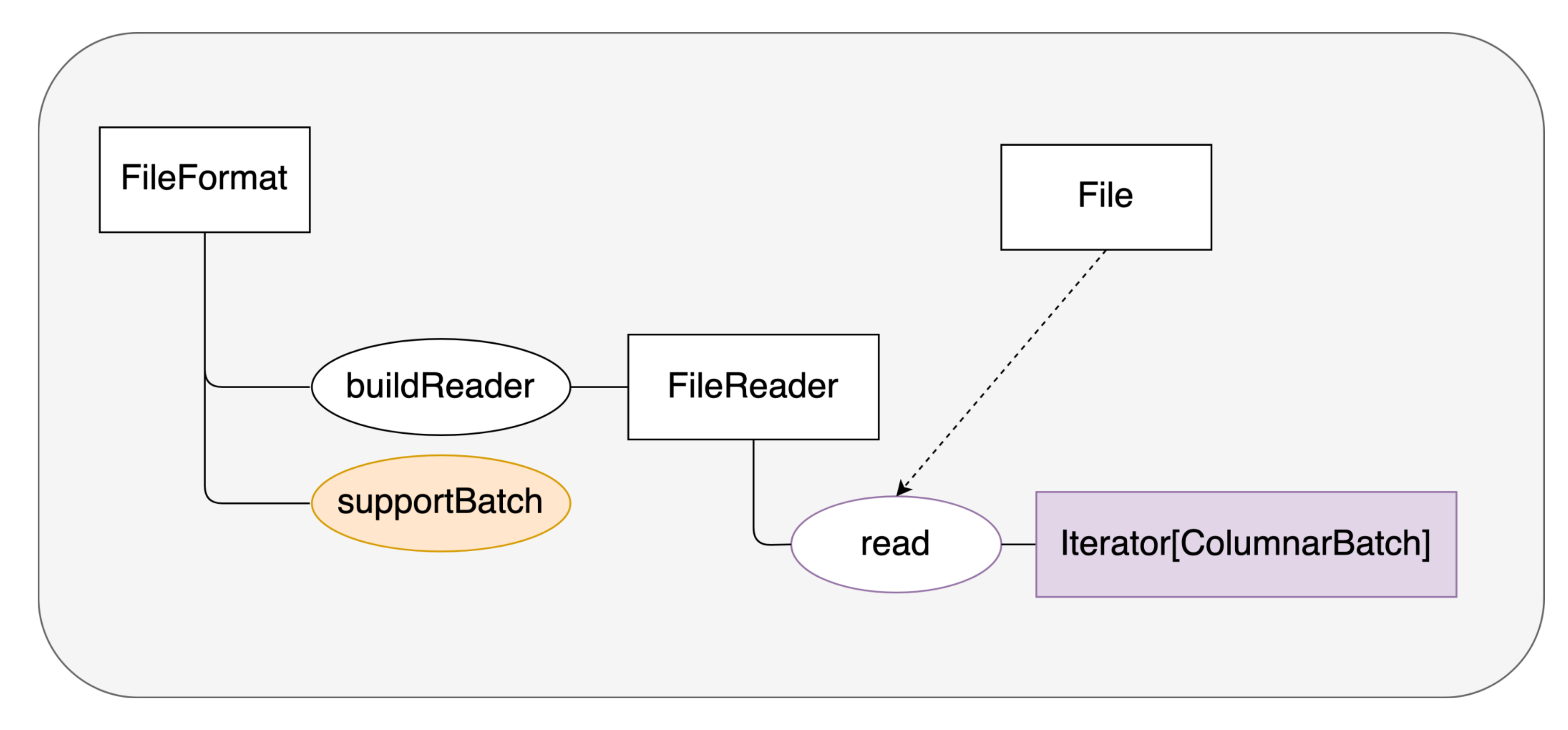
HL HighLoad++ Весна 2021

Построчное чтение



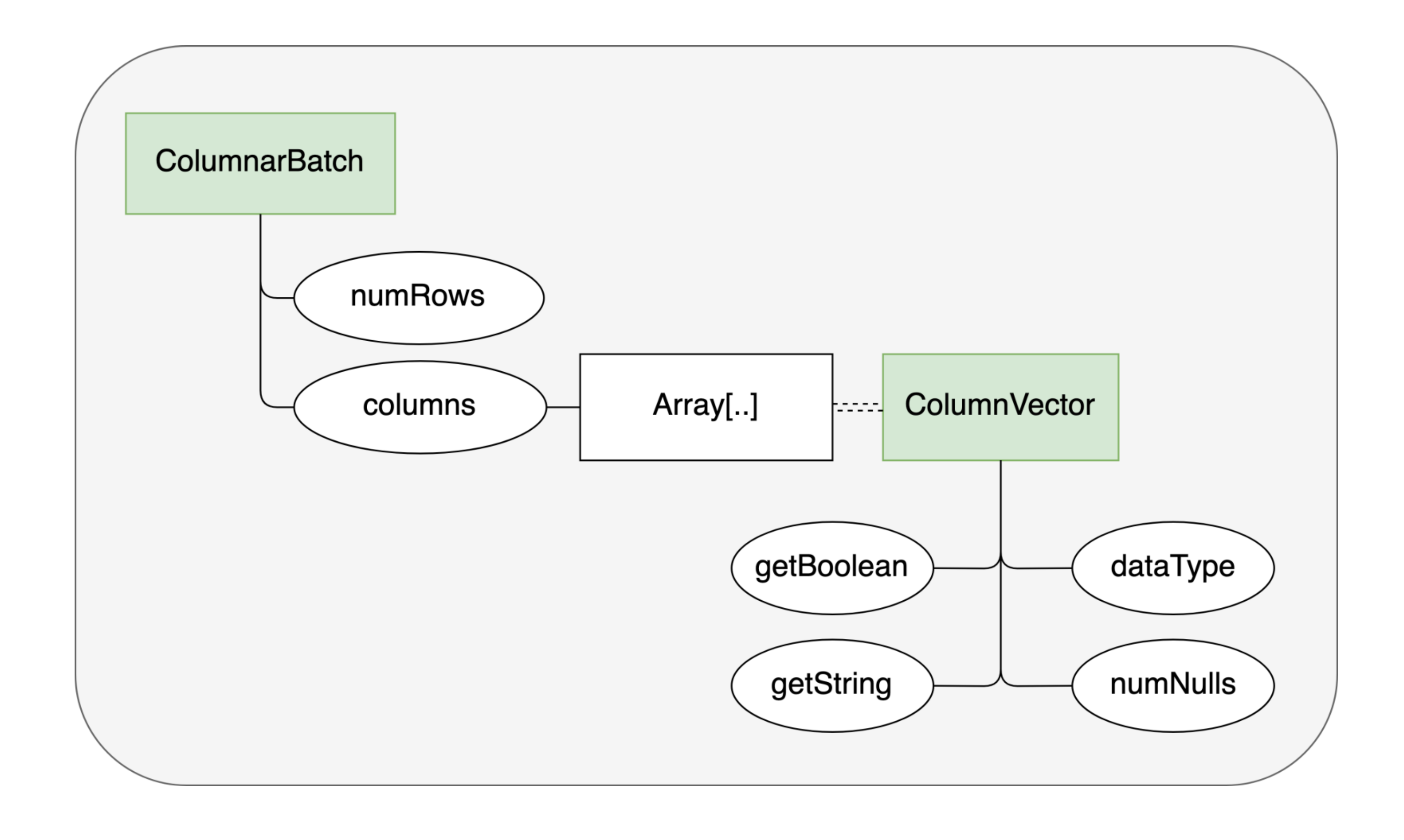
HL HighLoad++ Весна 2021

Чтение батчами



HL HighLoad++ Весна 2021

ColumnarBatch



HL HighLoad++ Весна 2021

Как Spark пишет Parquet в HDFS

HL HighLoad++ Весна 2021

Запись в Parquet на HDFS

```
/
|---> tmp
      |---> example
```

```
df.write.parquet("/tmp/example/dataset_1")
```


HL HighLoad++ Весна 2021

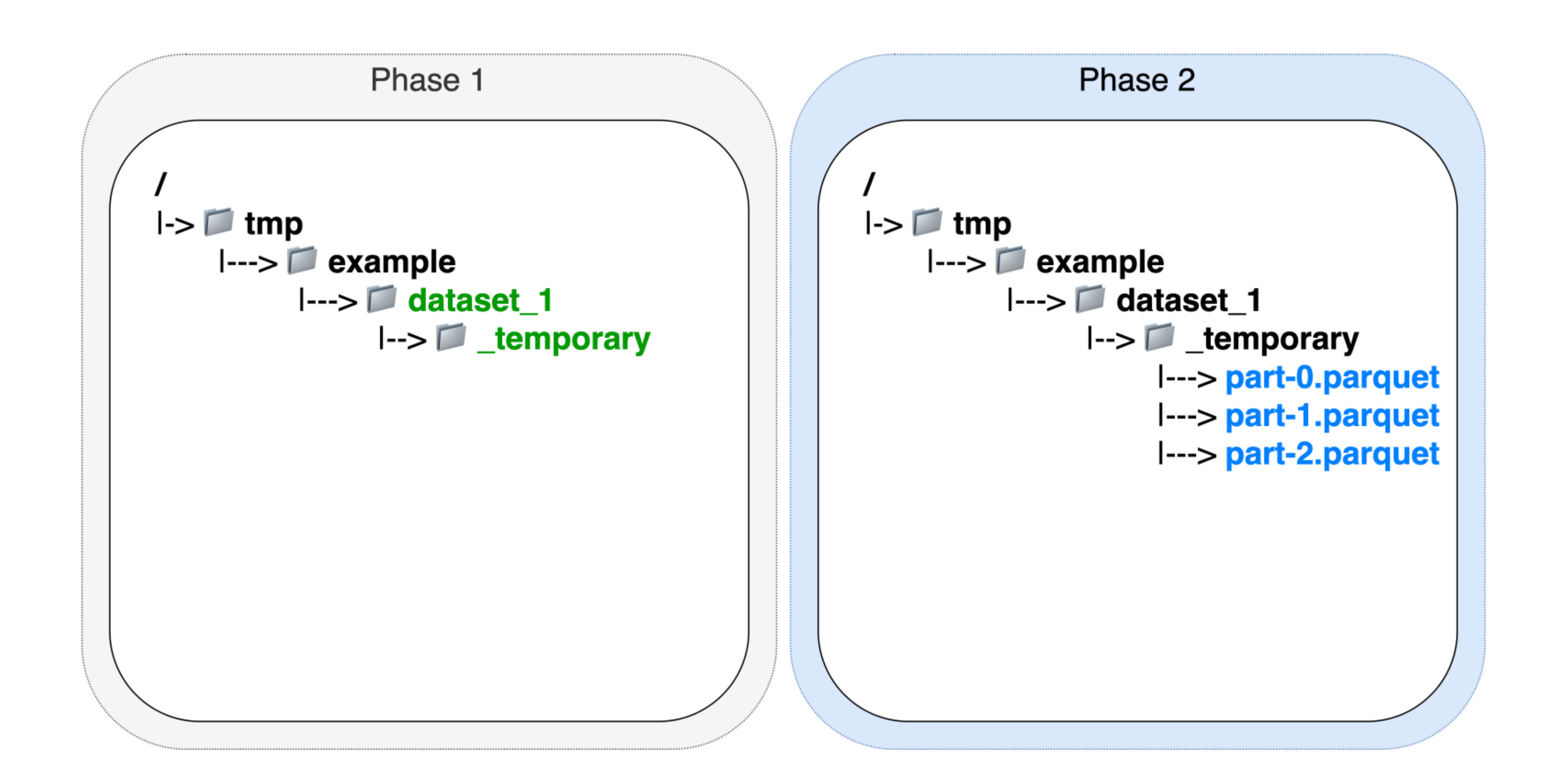
Создаём временную директорию

Phase 1

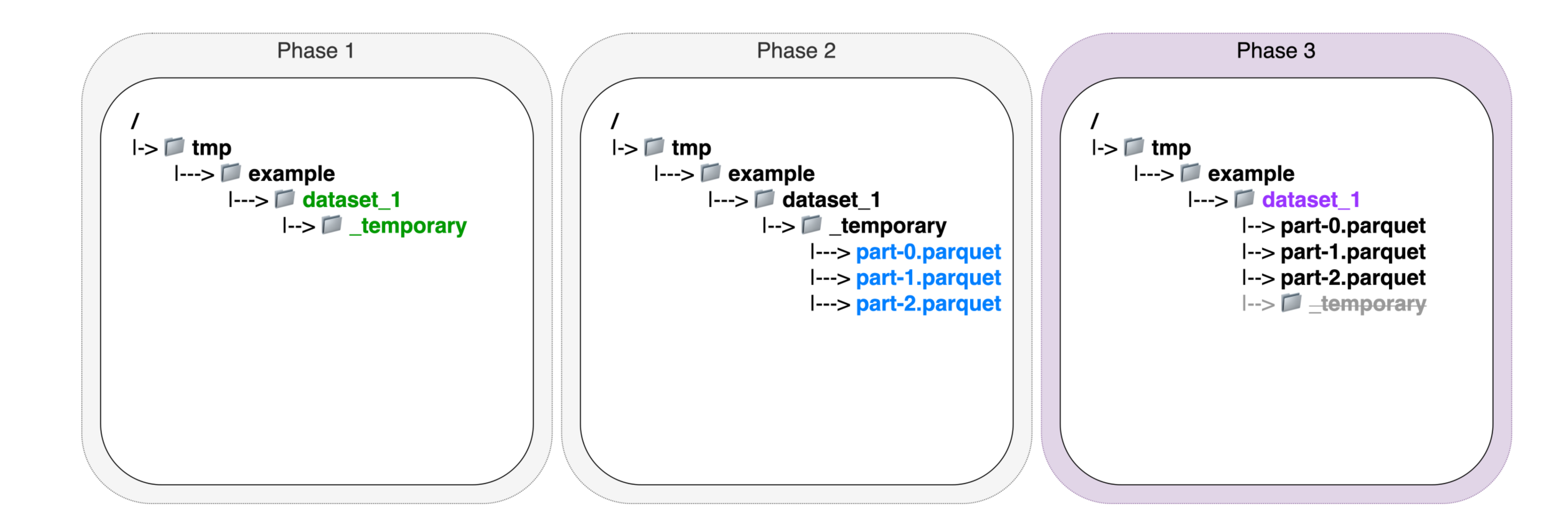
```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> _temporary
```


HL HighLoad++ Весна 2021

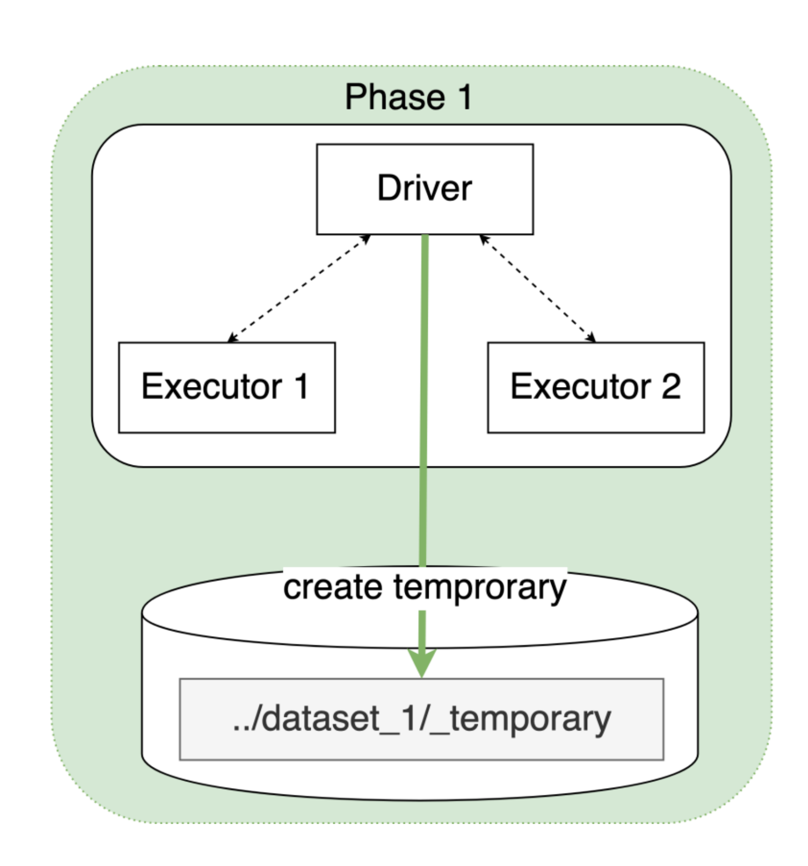
Пишем файлы во временную директорию



Перемещаем файлы

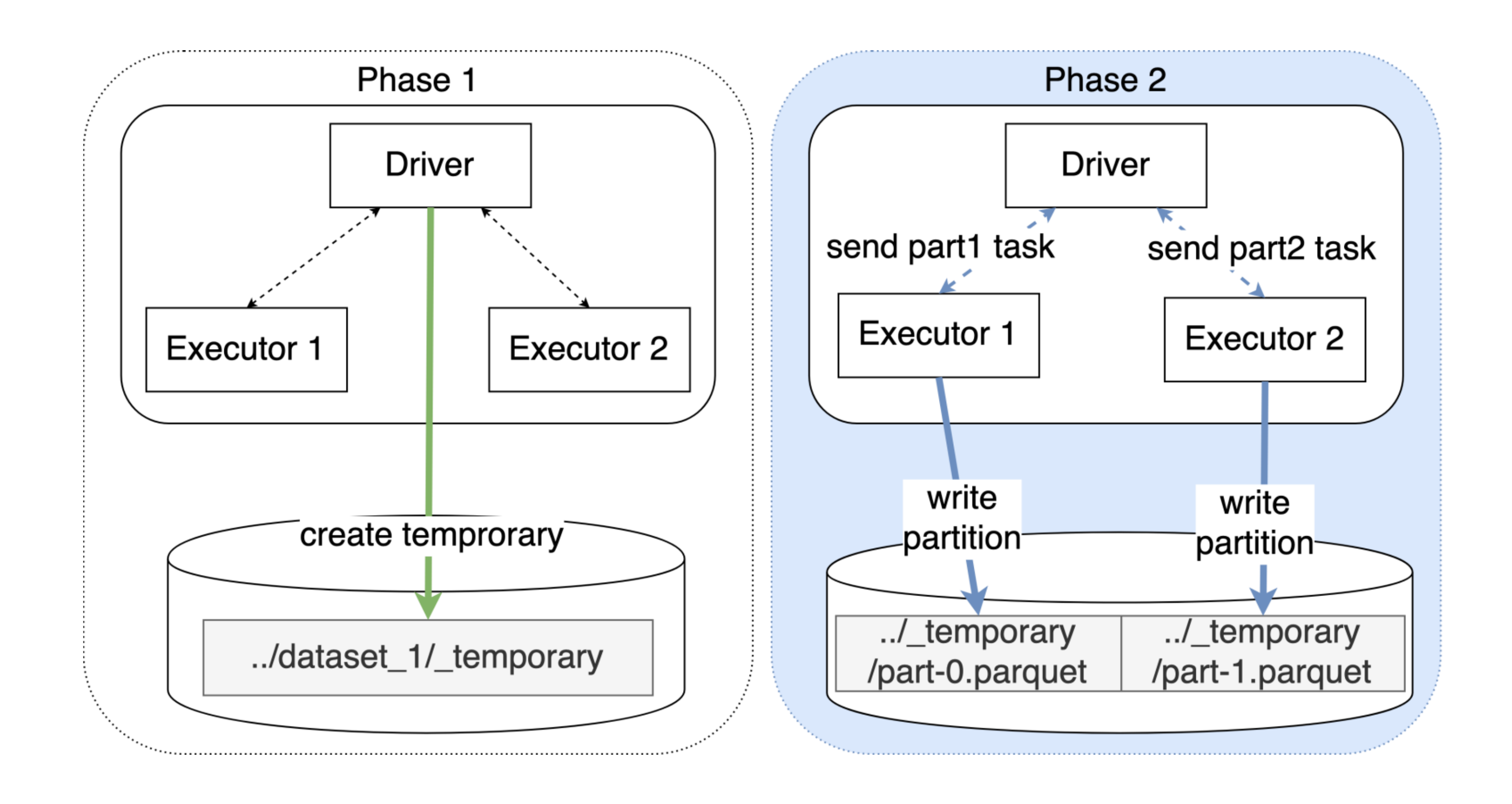


Создаём временную директорию

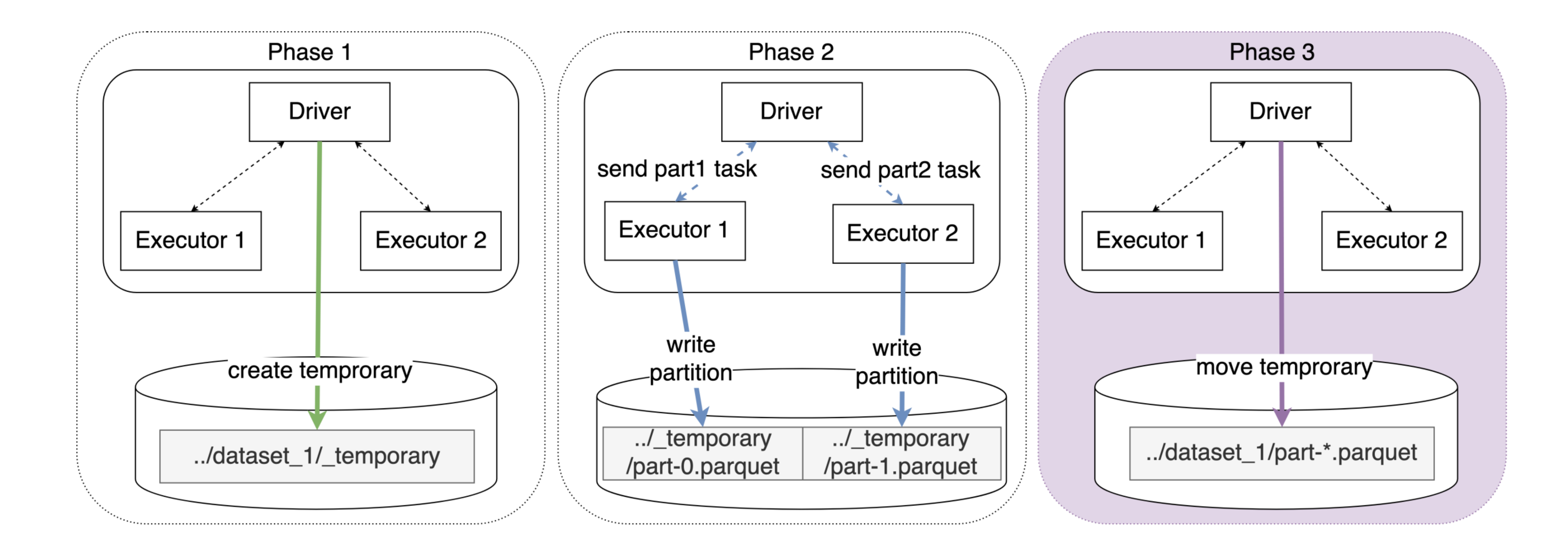


HL HighLoad++ Весна 2021

Пишем файлы во временную директорию

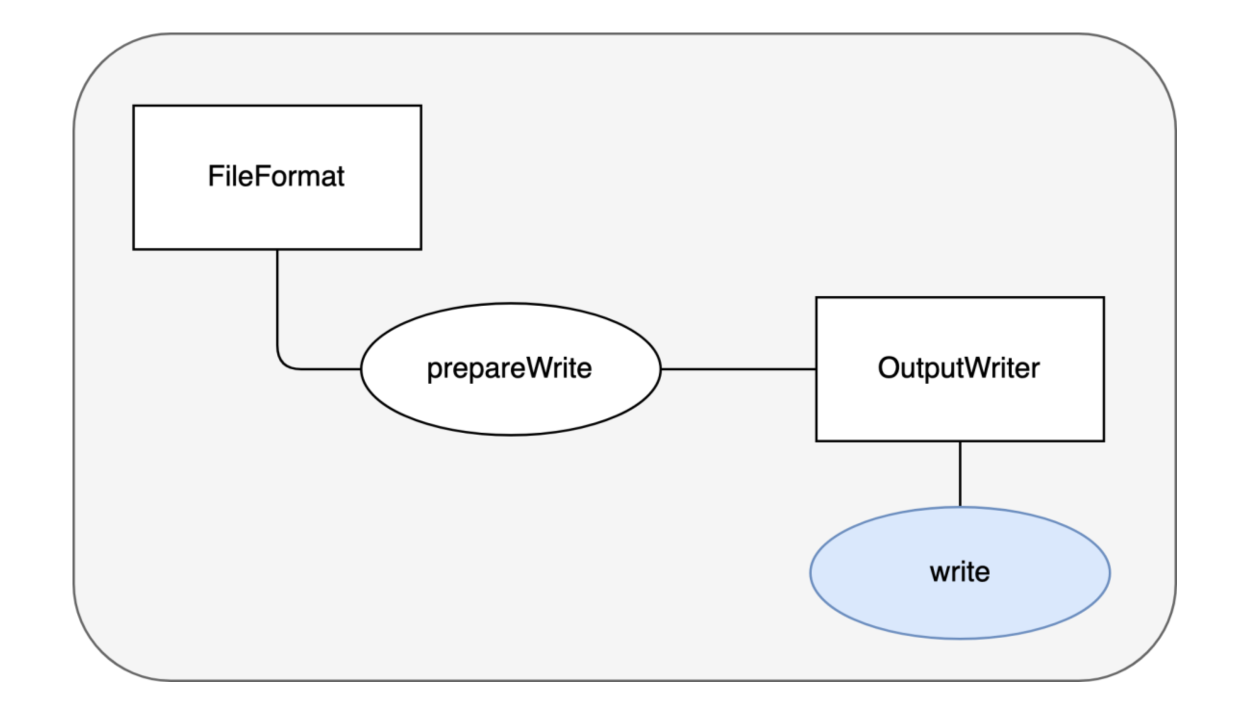


Перемещаем файлы



HL HighLoad++ Весна 2021

FileFormat

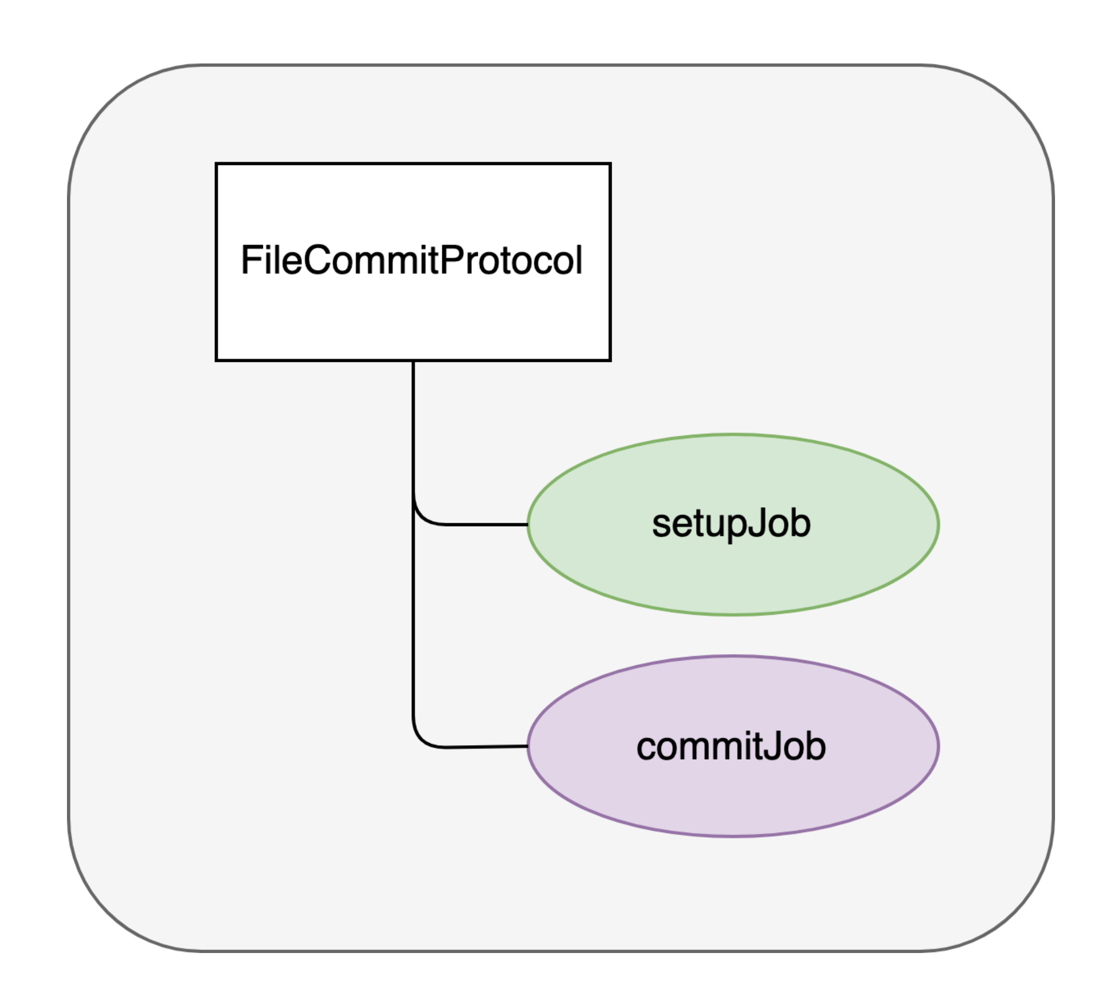



```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> _temporary
```

```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> _temporary
                     |---> part-0.parquet
                     |---> part-1.parquet
                     |---> part-2.parquet
```

```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> part-0.parquet
                |--> part-1.parquet
                |--> part-2.parquet
                |--> ~~_temporary~~
```

FileCommitProtocol

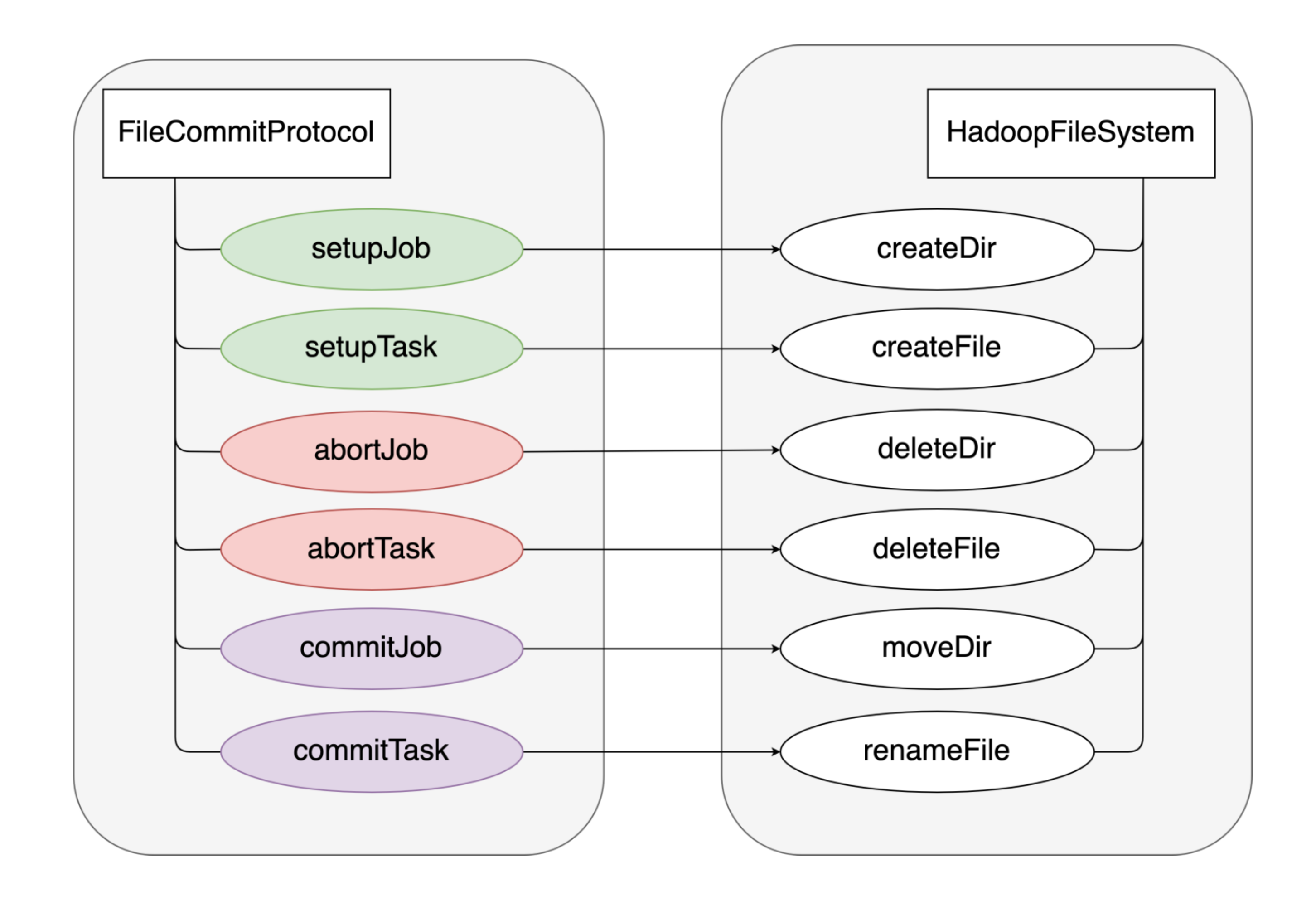



```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> _temporary
```

```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> _temporary
                     |---> part-0.parquet
                     |---> part-1.parquet
                     |---> part-2.parquet
```

```
/
|-> tmp
    |---> example
          |---> dataset_1
                |--> part-0.parquet
                |--> part-1.parquet
                |--> part-2.parquet
                |--> ~~_temporary~~
```

Эмулируем транзакции в HDFS

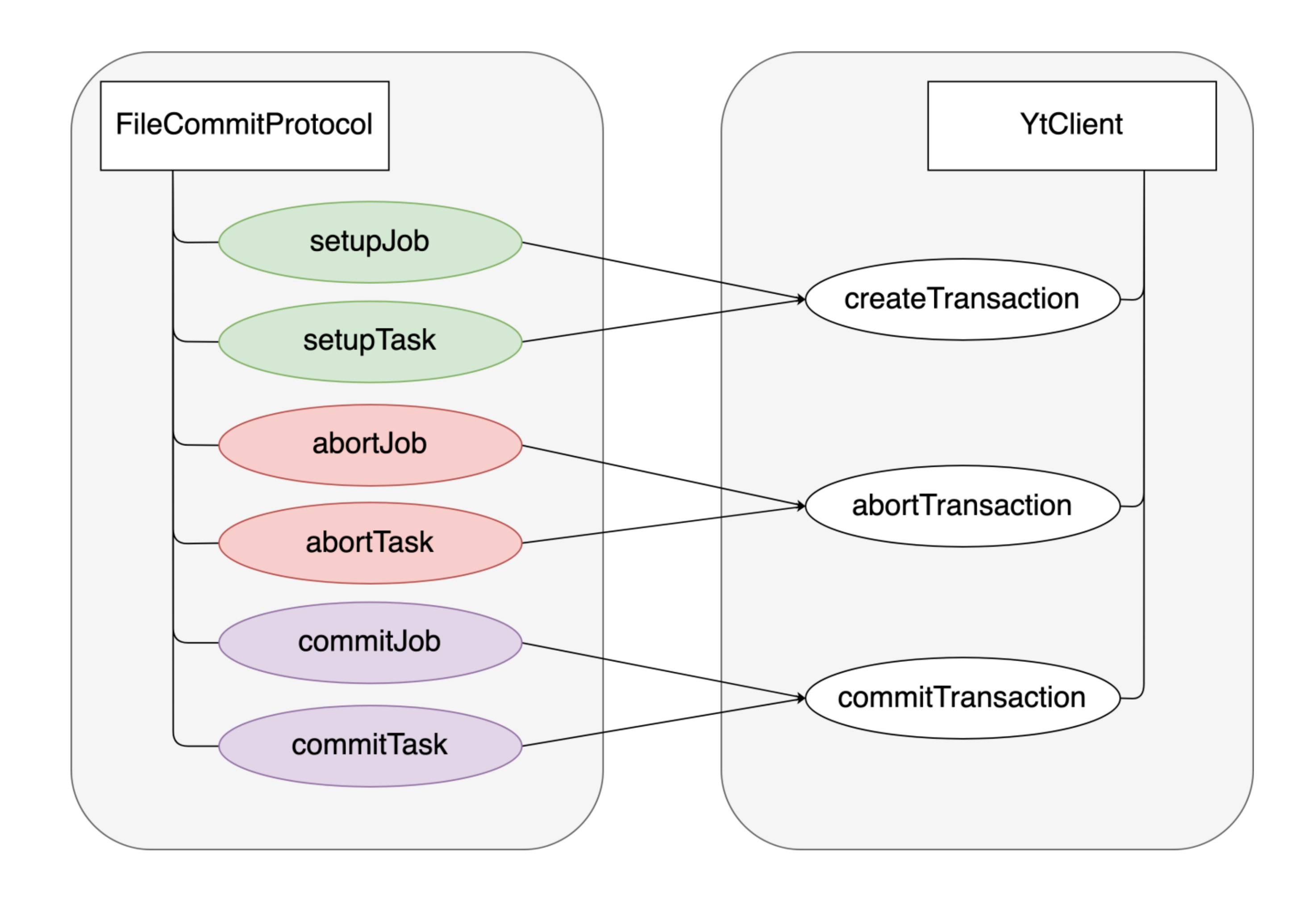


HL HighLoad++ Весна 2021

Как SPYT пишет таблицы в YT

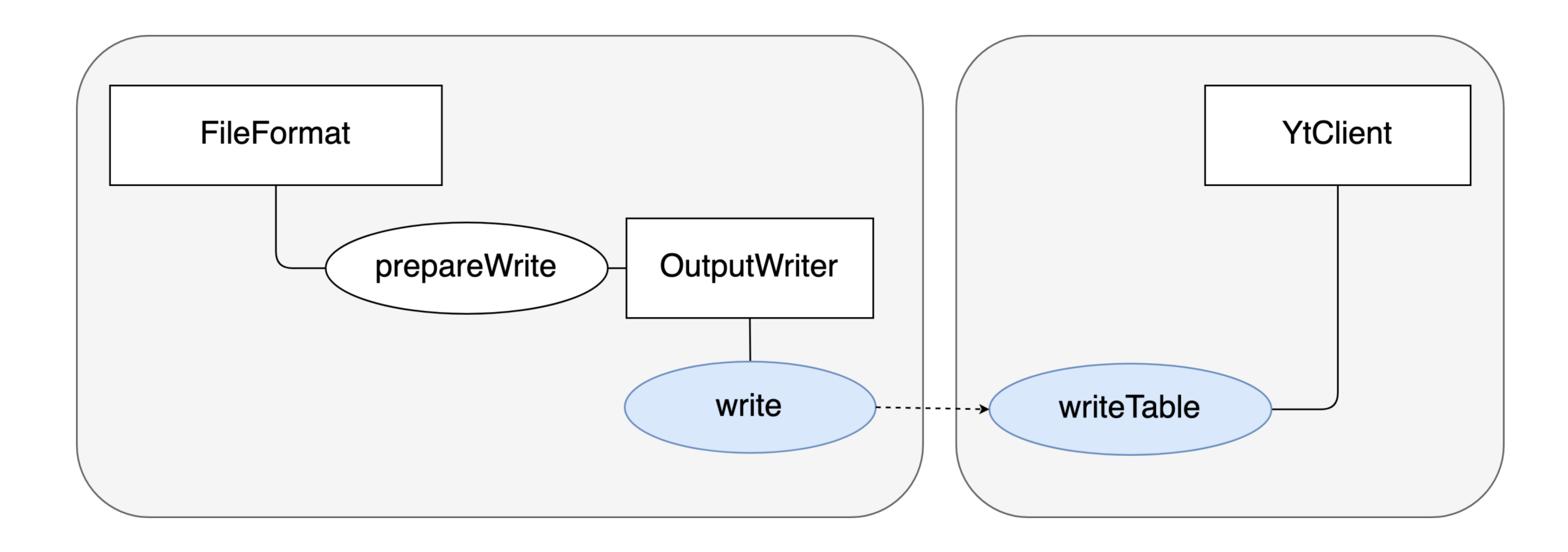
HL HighLoad++ Весна 2021

В YT уже есть транзакции!



HL HighLoad++ Весна 2021

Запись партиций в YT



HL HighLoad++ Весна 2021

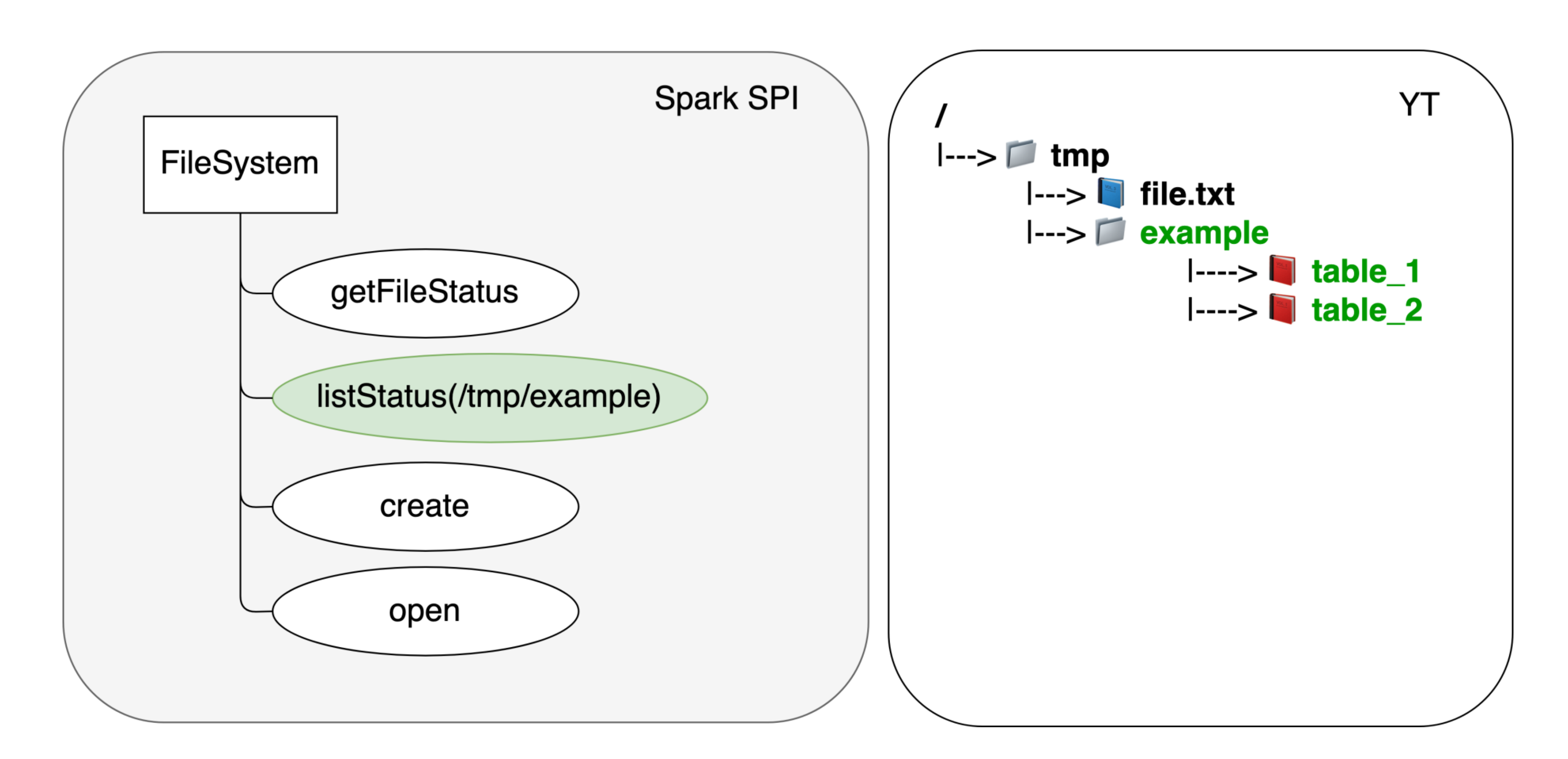
Итоги продвинутого подхода

- ☑ Научиться читать поколоночными батчами
- ☑ Переиспользовать код, написанный для HDFS и Parquet:
  - Partition discovery
  - Обход директорий на чтении
  - Создание / удаление временных файлов на записи

HL HighLoad++ Весна 2021

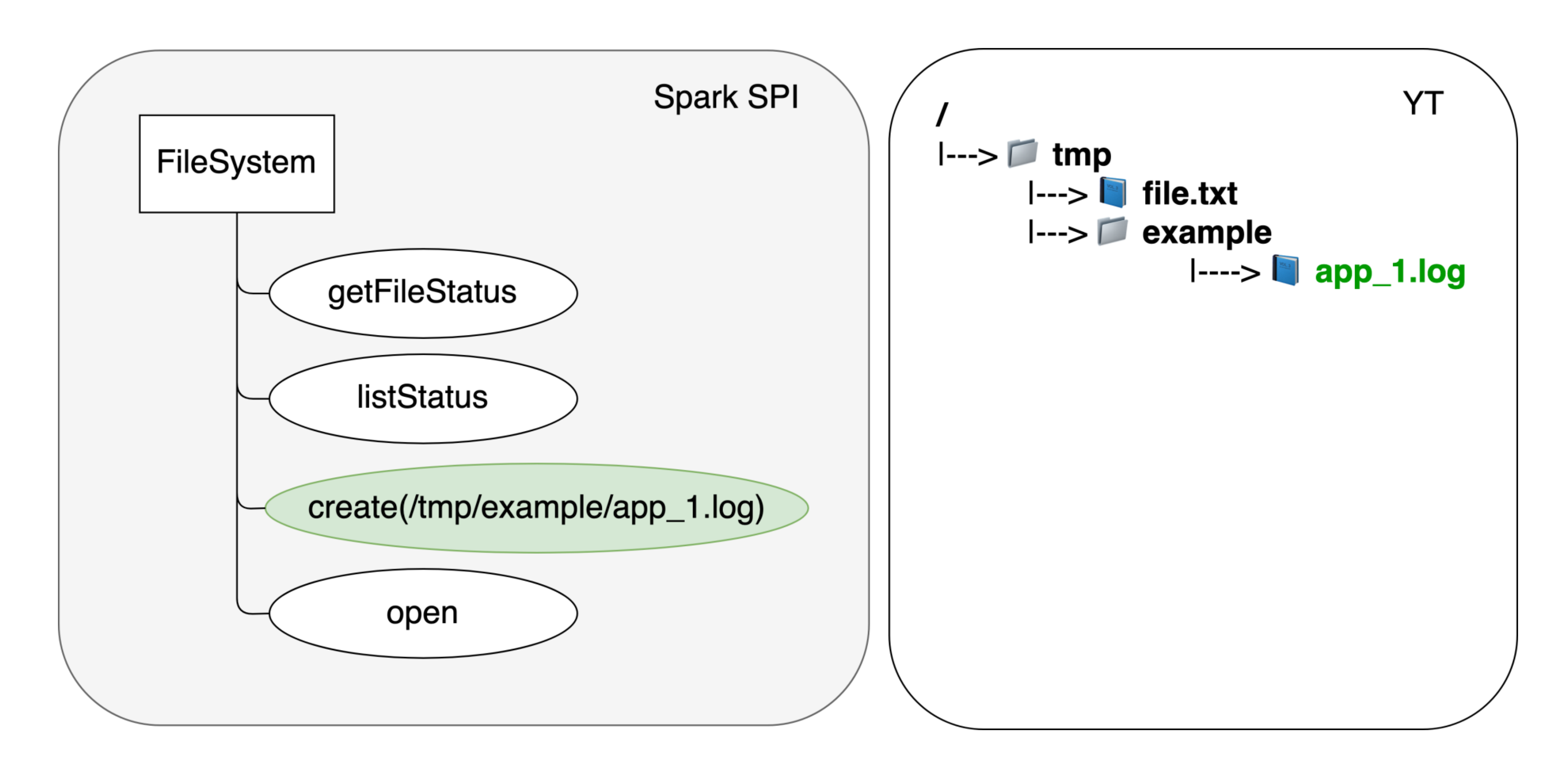
Дополнительные фичи

FileSystem



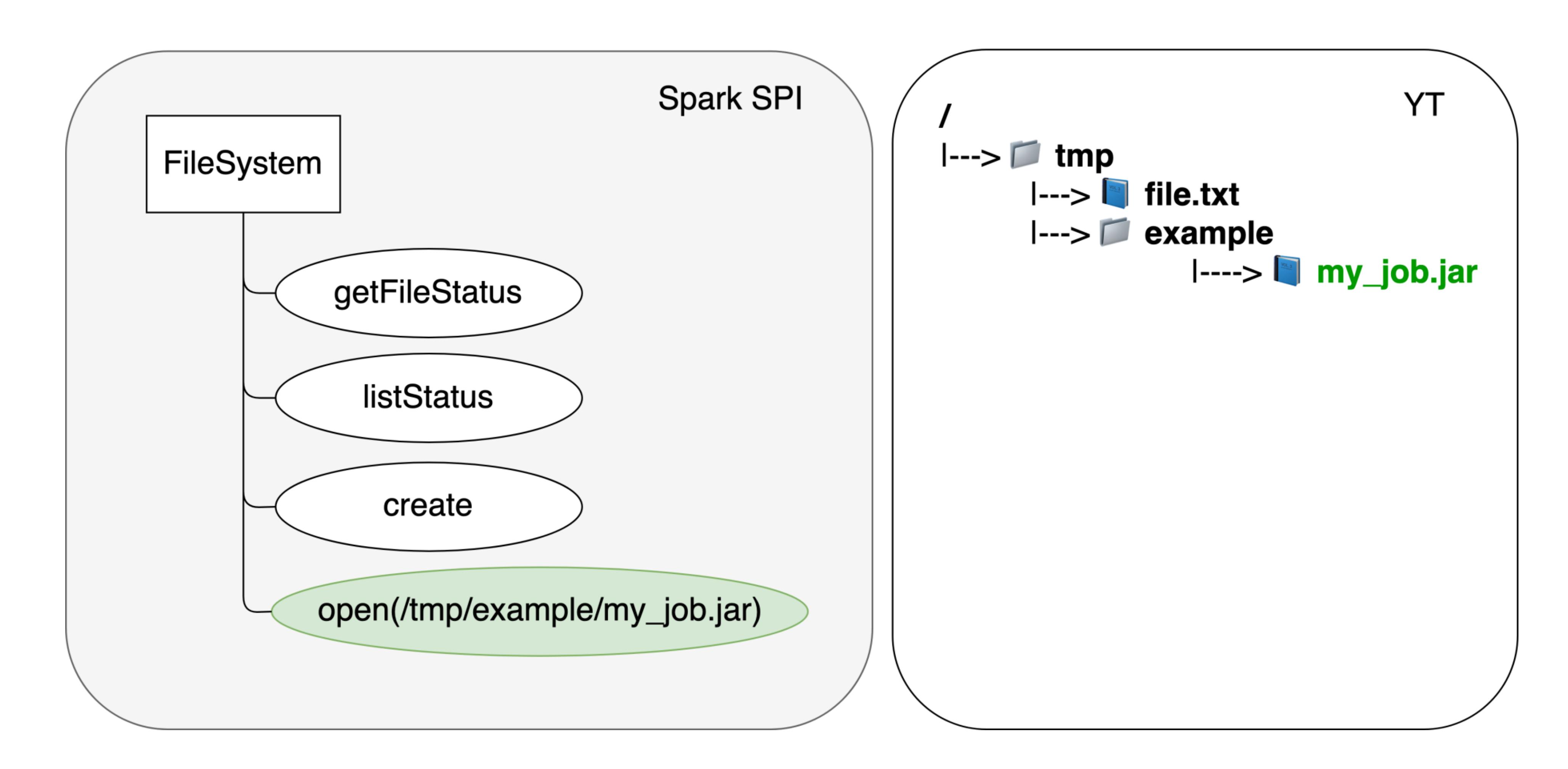
HL HighLoad++ Весна 2021

Создание файлов в YT



HL HighLoad++ Весна 2021

Чтение файлов из YT



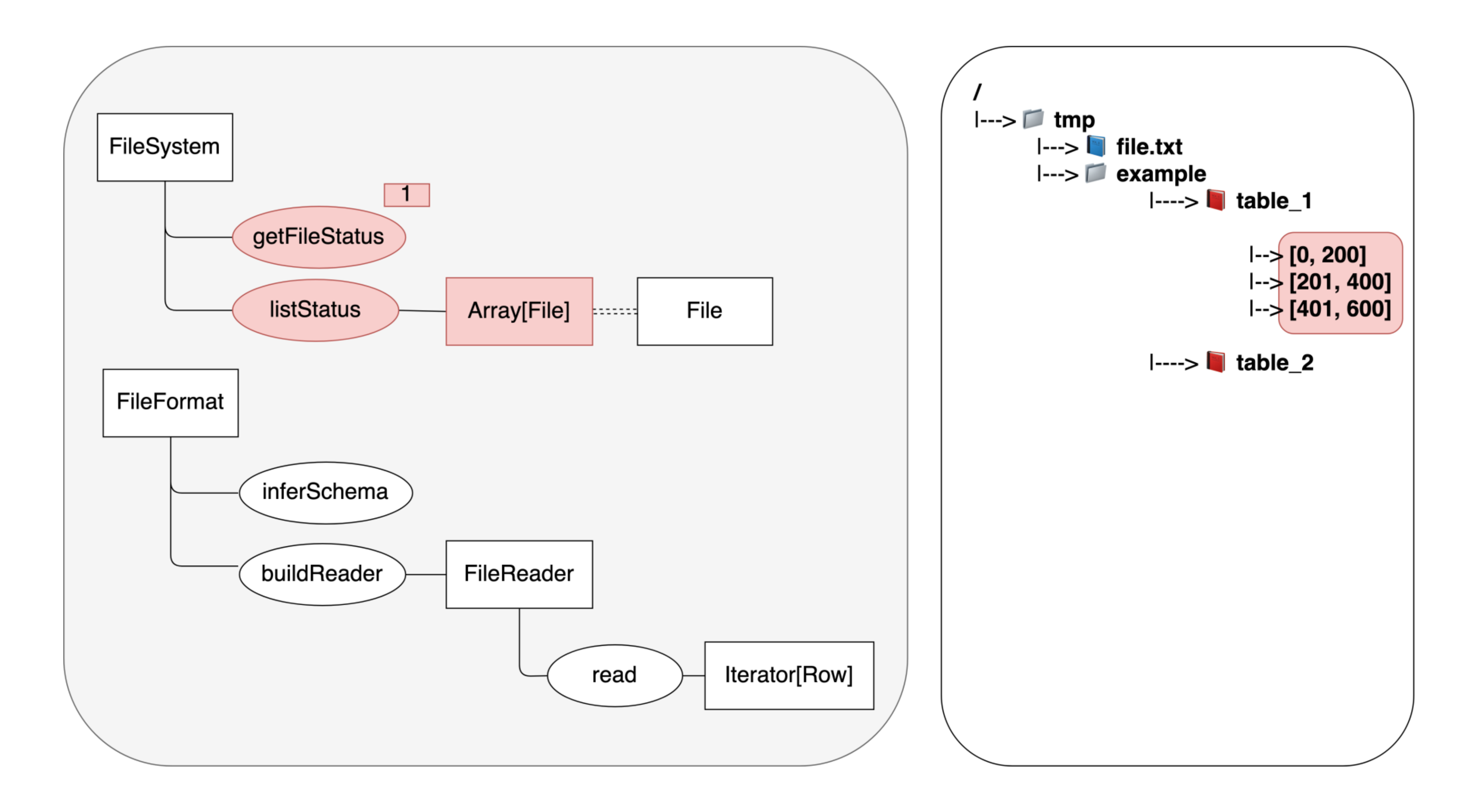
HL HighLoad++ Весна 2021

Для чего это?

- Заработало скачивание файлов из YT в spark-submit
- Есть возможность сохранять event log в YT и читать его в SparkHistoryServer

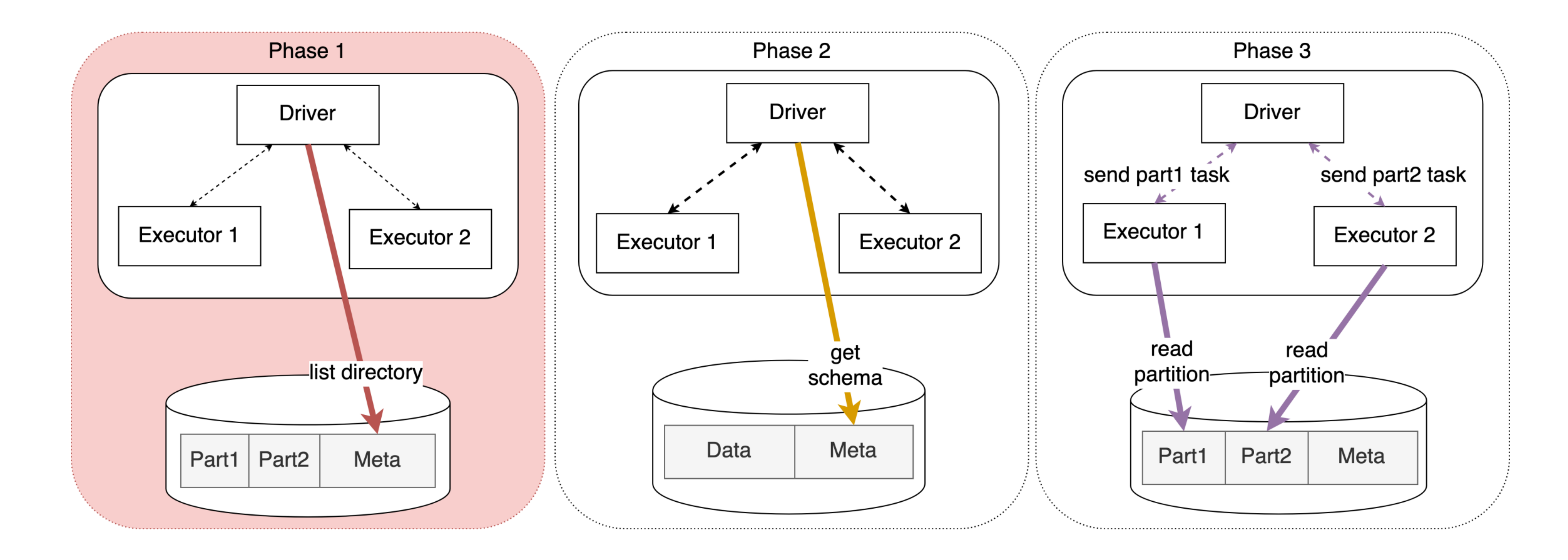
Грабли и костыли

Листинг таблицы при чтении



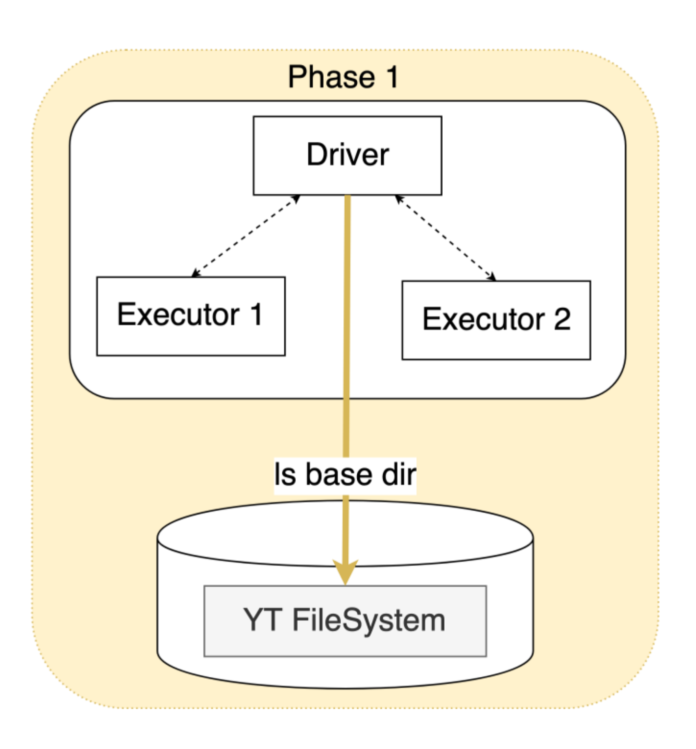
HL HighLoad++ Весна 2021

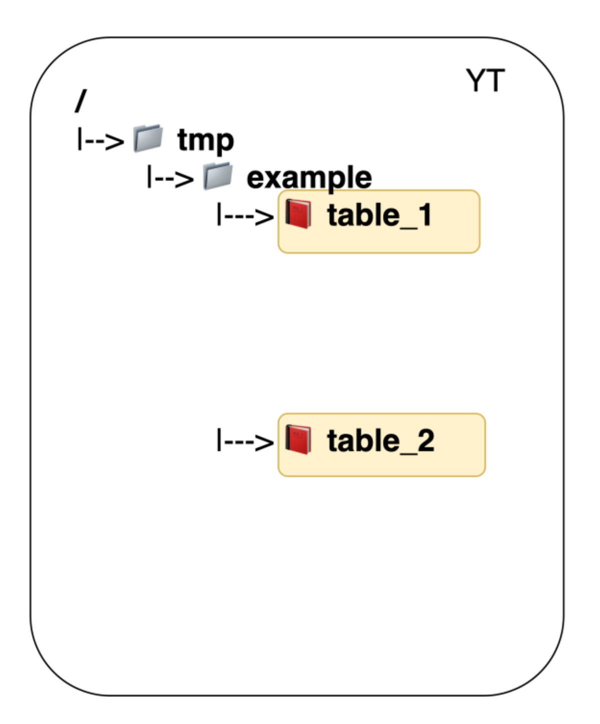
Листинг таблицы делает драйвер



HL HighLoad++ Весна 2021

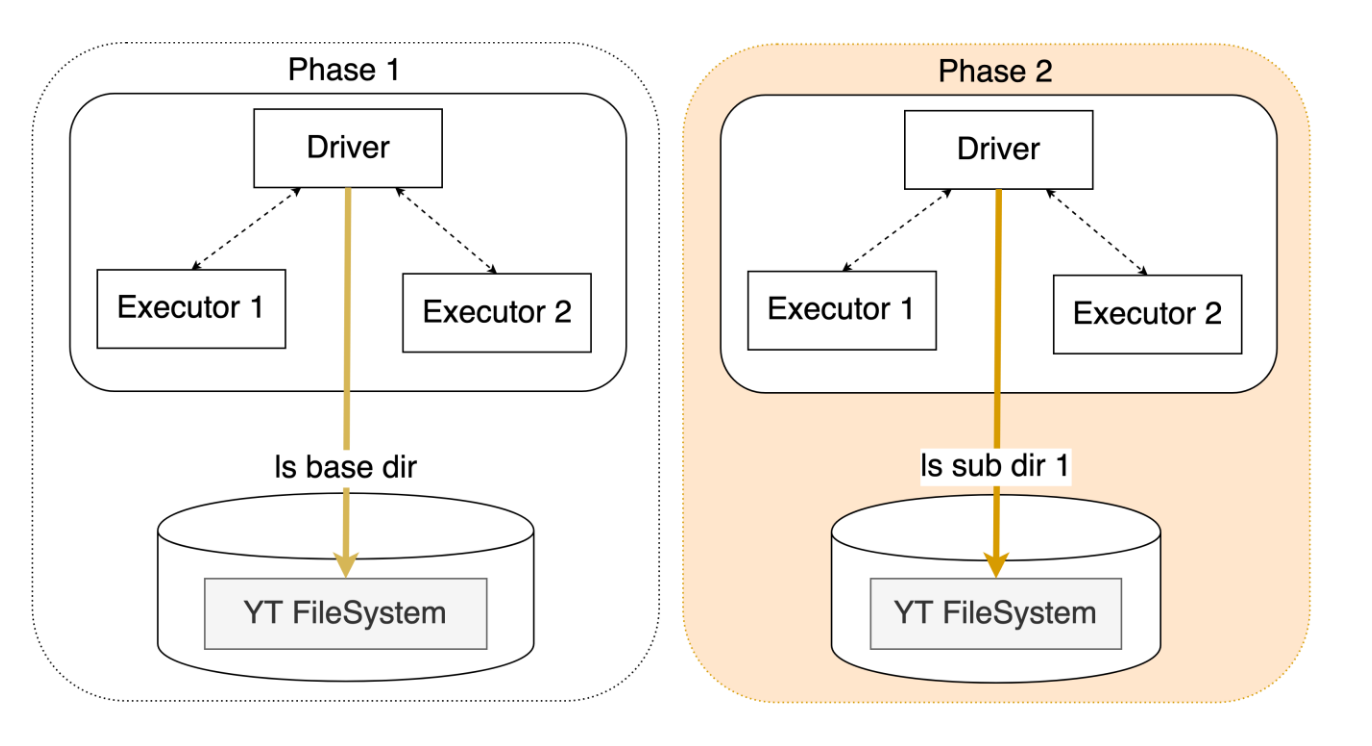
Читаем /tmp/example целиком

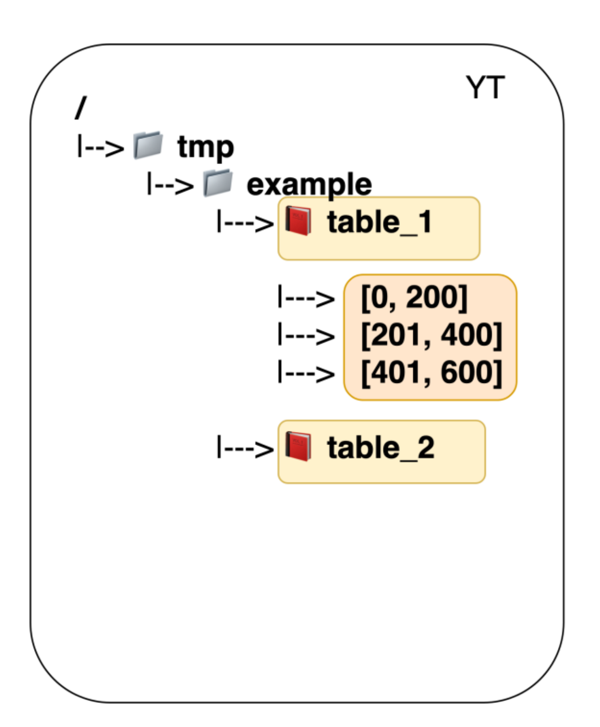




HL HighLoad++ Весна 2021

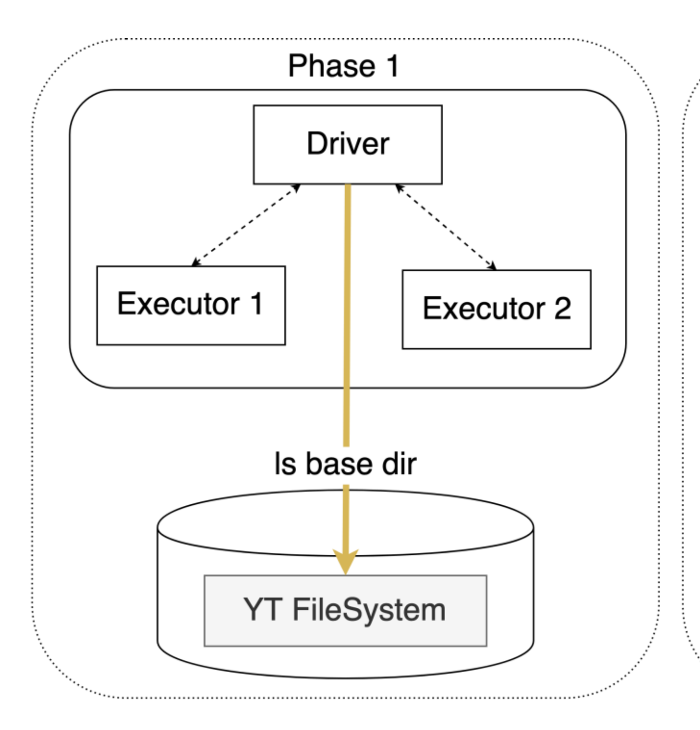
Читаем /tmp/example целиком

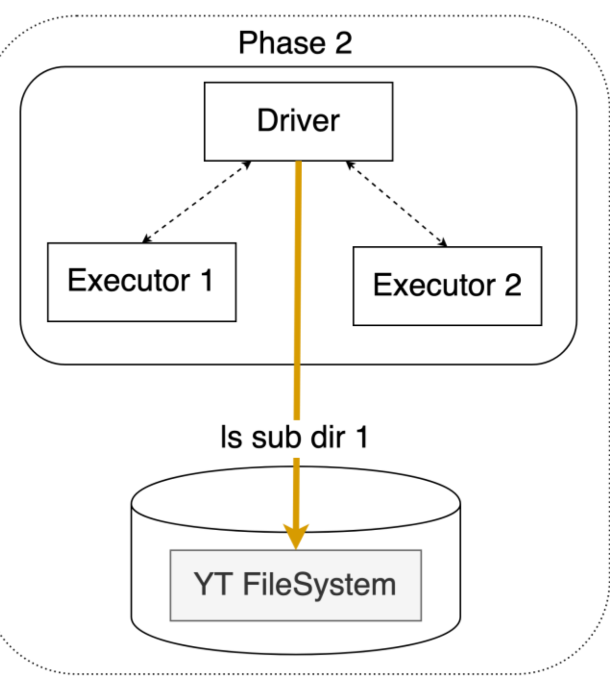


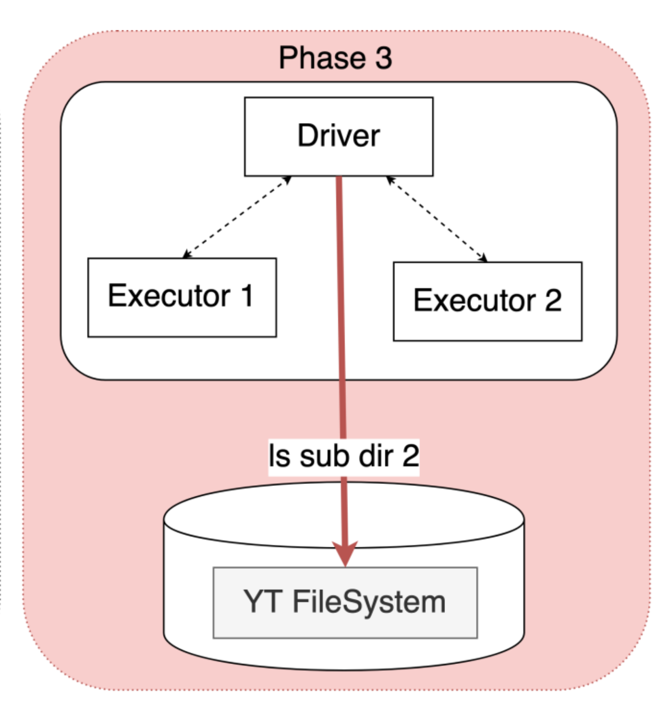


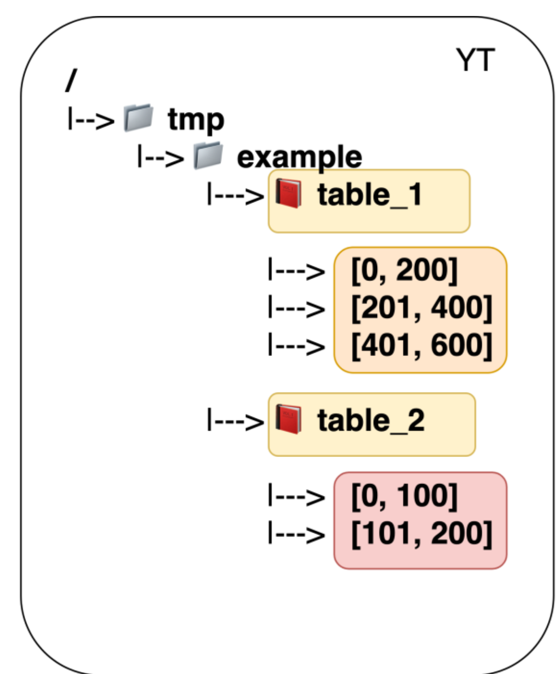
HL HighLoad++ Весна 2021

Однопоточный листинг партиций



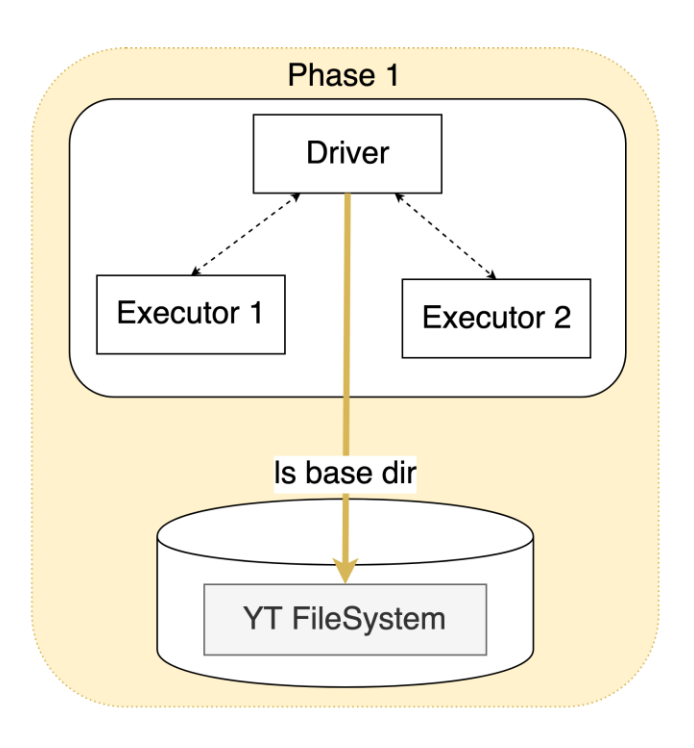


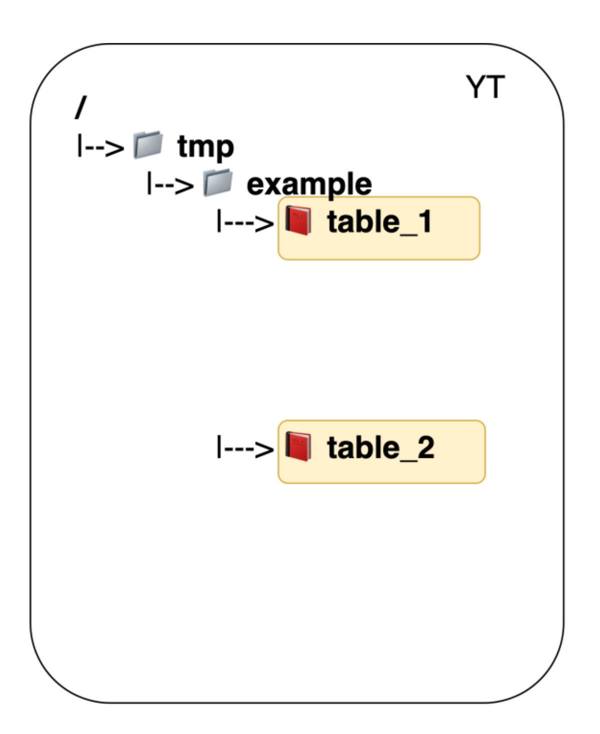




HL HighLoad++ Весна 2021

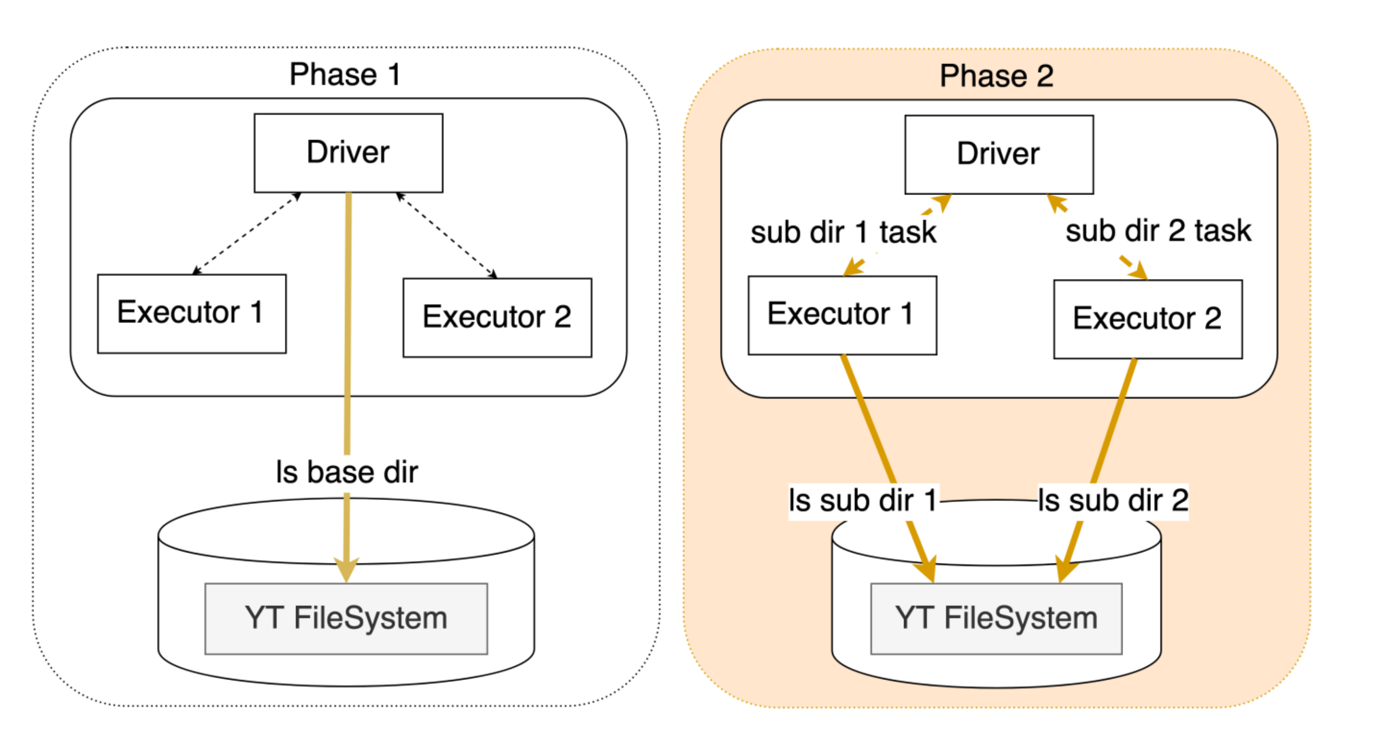
Читаем /tmp/example целиком

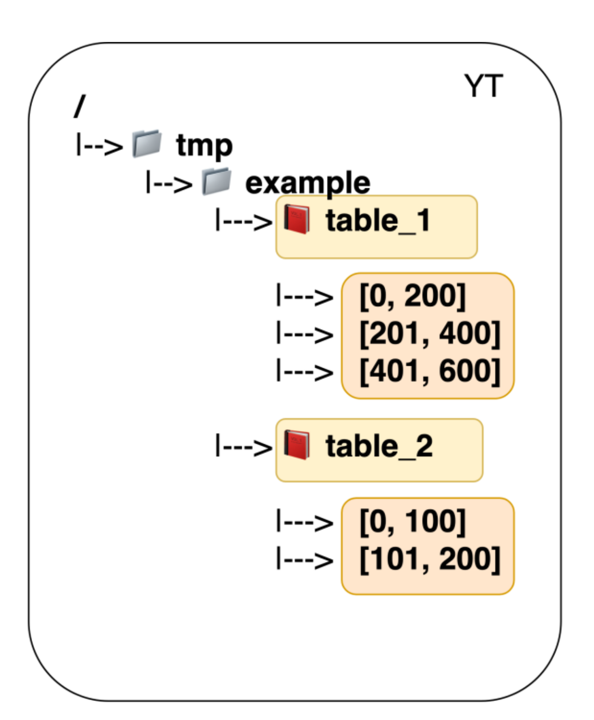




HL HighLoad++ Весна 2021

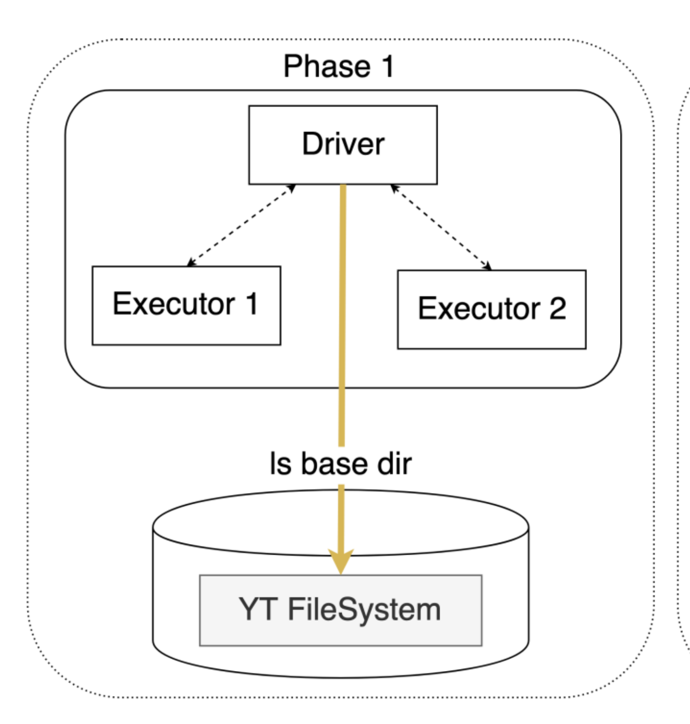
Параллельный листинг на экзекьюторах

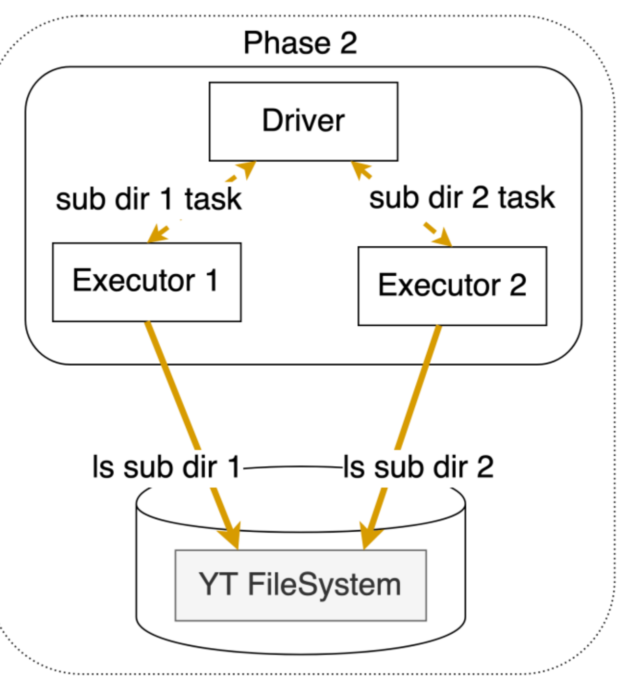


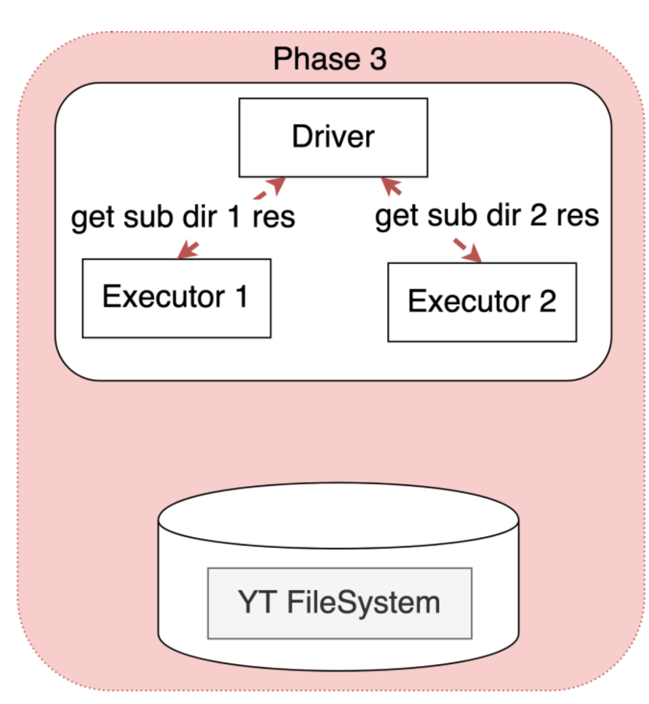


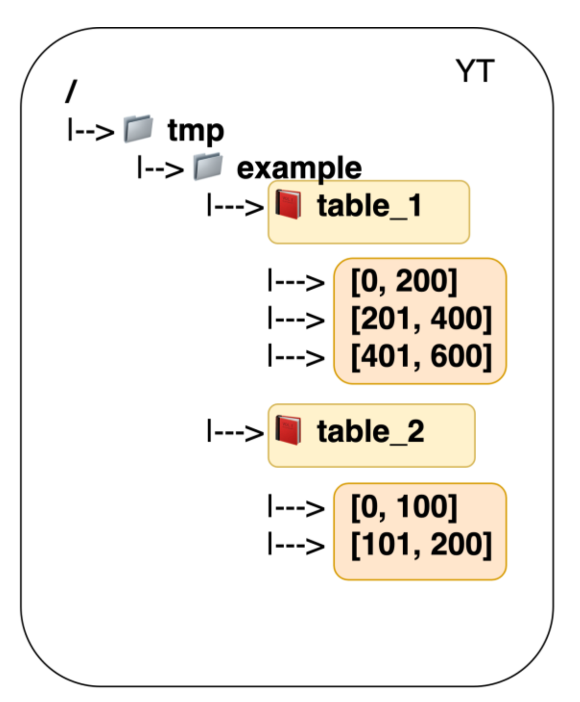
HL HighLoad++ Весна 2021

Parallel partition discovery



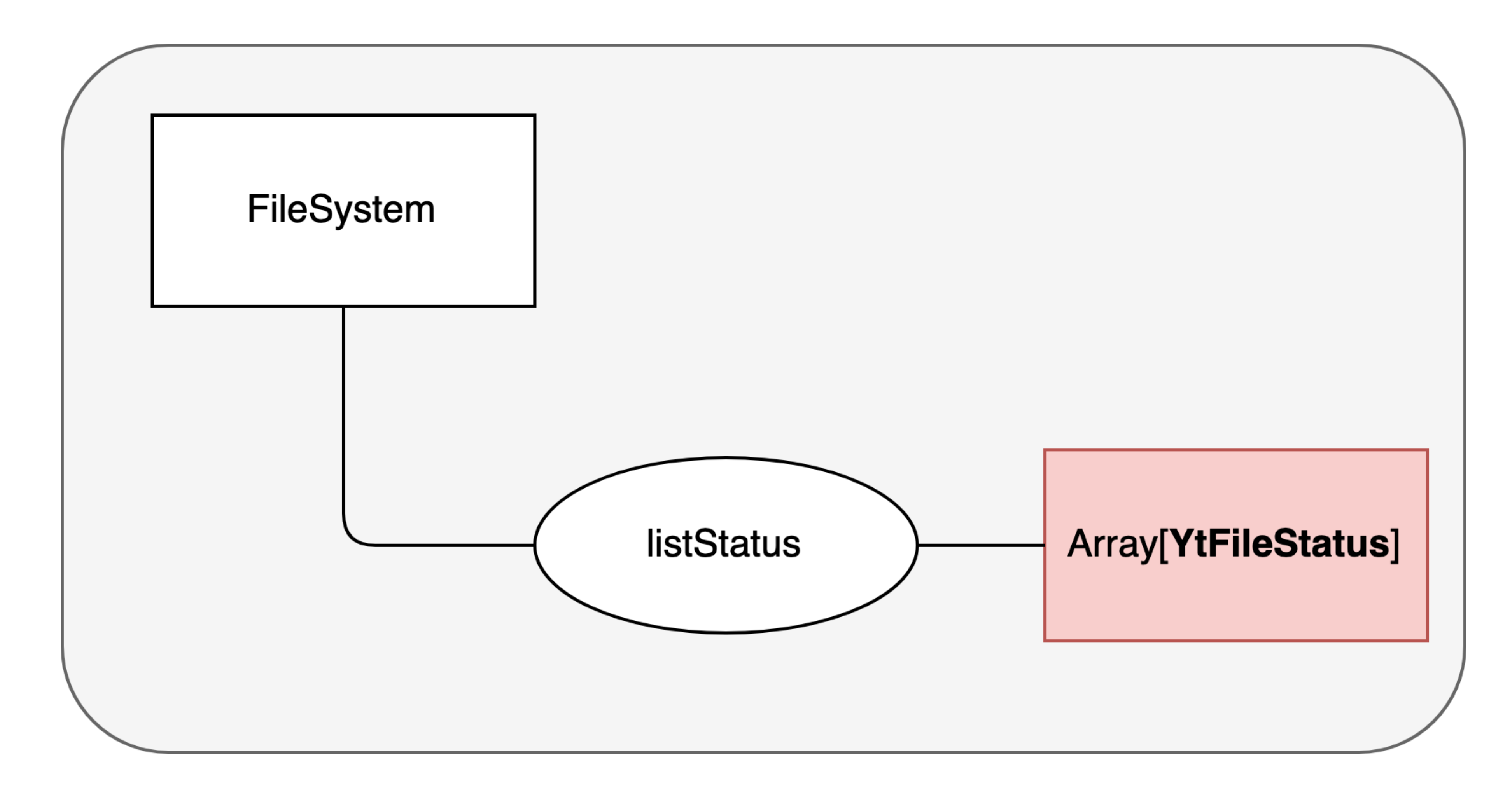






HL HighLoad++ Весна 2021

Реализация FileSystem



HL HighLoad++ Весна 2021

Что с этим не так?

listStatus вызывается на экзекьюторе

Что с этим не так?

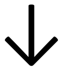
listStatus вызывается на экзекьюторе

На экзекьюторе получаем Array[YtFileStatus]

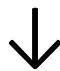
HL HighLoad++ Весна 2021

Что с этим не так?

listStatus вызывается на экзекьюторе



На экзекьюторе получаем Array[YtFileStatus]

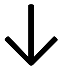


Сериализуем и отправляем на драйвер

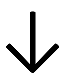
HL HighLoad++ Весна 2021

Что с этим не так?

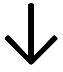
listStatus вызывается на экзекьюторе



На экзекьюторе получаем Array[YtFileStatus]



Сериализуем и отправляем на драйвер



Десериализуем на драйвере

HL HighLoad++ Весна 2021

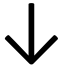
Что с этим не так?

listStatus вызывается на экзекьюторе

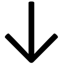
↓

На экзекьюторе получаем Array[YtFileStatus]

Сериализуем и отправляем на драйвер



Десериализуем на драйвере



Array[FileStatus]

Что делать?

- Учитывать, что методы для листинга директории могут вызываться на экзекьюторах
- Не пытаться обмануть SPI и использовать FileStatus

HL HighLoad++ Весна 2021

FileSystem – синглтон?

```
public static FileSystem get(URI uri,
                             Configuration conf)
{
    …
    return CACHE.get(uri, conf);
}
```


HL HighLoad++ Весна 2021

Использую синглтон в init / close

```
class YtFileSystem extends FileSystem {
  override def initialize(…): Unit = {
    …
    initYtClient()
  }
  override def close(): Unit = {
    …
    closeYtClient()
  }
  …
}
```


HL HighLoad++ Весна 2021

Как устроен кэш FileSystem

```
private FileSystem getInternal ( Key key, …) {
    …
    synchronized (this) {
        fs = map.get(key);
    }
    if (fs != null) return fs;

    fs = createFileSystem(uri, conf);

    synchronized (this) {               // refetch the lock again
        FileSystem oldfs = map.get(key);
        if (oldfs != null) {            // FS created while lock is releasing
            fs.close();                 // close the new file system
            return oldfs;               // return the old file system
        }

        …
        return fs;
    }
}
```


HL HighLoad++ Весна 2021

Пытаемся взять FS из кэша

```
private FileSystem getInternal ( Key key, …) {
    …
    synchronized (this) {
        fs = map.get(key);
    }
    if (fs != null) return fs;

    fs = createFileSystem(uri, conf);

    synchronized (this) {                    // refetch the lock again
        FileSystem oldfs = map.get(key);
        if (oldfs != null) {                 // FS created while lock is releasing
            fs.close();                      // close the new file system
            return oldfs;                    // return the old file system
        }

        …
        return fs;
    }
}
```


HL HighLoad++ Весна 2021

Создаём новый инстанс FS

```
private FileSystem getInternal ( Key key, …) {
    …
    synchronized (this) {
        fs = map.get(key);
    }
    if (fs != null) return fs;

    fs = createFileSystem(uri, conf);

    synchronized (this) {                 // refetch the lock again
        FileSystem oldfs = map.get(key);
        if (oldfs != null) {              // FS created while lock is releasing
            fs.close();                   // close the new file system
            return oldfs;                 // return the old file system
        }

        …
        return fs;
    }
}
```


HL HighLoad++ Весна 2021

Другой поток создал инстанс раньше

```
private FileSystem getInternal ( Key key, …) {
    …
    synchronized (this) {
        fs = map.get(key);
    }
    if (fs != null) return fs;

    fs = createFileSystem(uri, conf);

    synchronized (this) {                    // refetch the lock again
        FileSystem oldfs = map.get(key);
        if (oldfs != null) {                 // FS created while lock is releasing
            fs.close();                      // close the new file system
            return oldfs;                    // return the old file system
        }

        …
        return fs;
    }
}
```


HL HighLoad++ Весна 2021

Закрываем свой инстанс

```
private FileSystem getInternal ( Key key, …) {
     …
     synchronized (this) {
          fs = map.get(key);
     }
     if (fs != null) return fs;

     fs = createFileSystem(uri, conf);

     synchronized (this) {                   // refetch the lock again
          FileSystem oldfs = map.get(key);
          if (oldfs != null) {               // FS created while lock is releasing
               fs.close();                   // close the new file system
               return oldfs;                 // return the old file system
          }

          …
          return fs;
     }
}
```


HL HighLoad++ Весна 2021

Что делать?

- Запомнить, что FileSystem может быть создан не один, а лишний сразу будет закрыт

- Не использовать настоящие синглтоны в коде FileSystem

HL HighLoad++ Весна 2021

Кастомные параметры в Spark

- Большую часть параметров подключения можно прокидывать через конфиг спарка:
  `--conf spark.my.param=value`

- Их удобно доставать из SparkConf, можно определять дефолтные значения в `spark-defaults.conf`

- Но они видны в логах и на вкладке Environment в Spark UI

HL HighLoad++ Весна 2021

Как скрыть отображение секретных параметров?

- В Spark есть специальный конфиг

  `spark.redaction.regex`

- Все параметры, названия которых соответствуют регулярке, будут скрыты звёздочками

HL HighLoad++ Весна 2021

Как скрыть отображение секретных параметров?

- В Spark есть специальный конфиг
  `spark.redaction.regex`

- Все параметры, названия которых соответствуют регулярке, будут скрыты звёздочками

- Или нет?

HL HighLoad++ Весна 2021

Что я сделала

- Прокинула параметр

```
spark.yt.token=token
```

- Сказала спарку, что он секретный

```
spark.redaction.regex=(?i)secret|password|token
```


HL HighLoad++ Весна 2021

Что получилось

- [x] Параметр скрыт в SparkUI

HL HighLoad++ Весна 2021

Но в логах драйвера…

- Запуск в Spark Standalone логирует строку запуска драйвера полностью

```
Launch Command: "java" "-cp"
"-Dspark.yt.token=token" …
org.apache.spark.deploy.worker.DriverWrapper …
```


HL HighLoad++ Весна 2021

Что получилось

- ☑ Параметр скрыт в SparkUI
- ✕ Параметр виден в логах драйвера

HL HighLoad++ Весна 2021

Что делать?

- В Spark 3.0 это починили

```
Launch Command: "java" "-cp"
"-Dspark.yt.token=*********(redacted)" …
org.apache.spark.deploy.worker.DriverWrapper …
```


HL HighLoad++ Весна 2021

Что получилось

- ✓ Параметр скрыт в SparkUI
- ✓ Параметр скрыт в логах драйвера

HL HighLoad++ Весна 2021

Но в списке процессов…

- На хосте, где запущен spark-submit, видны все его аргументы

```
> ps aux | grep spark

spark-submit … --conf spark.yt.token=token …
```


HL HighLoad++ Весна 2021

Может, обойти spark-submit?

- Всё равно придём к запуску класса SparkSubmit

```
> ps aux | grep spark

java -cp … org.apache.spark.deploy.SparkSubmit
  --conf spark.yt.token=token …
```

Что получилось

- ☑ Параметр скрыт в SparkUI
- ☑ Параметр скрыт в логах драйвера
- ✕ Параметр виден в ps на хосте запуска spark-submit

HL HighLoad++ Весна 2021

Что делать?

- Костыль в SparkSubmit.scala

```
sys.env.get("SPARK_YT_TOKEN").foreach {
    token =>
        sparkConf.set("spark.yt.token", token)
}
```


HL HighLoad++ Весна 2021

Что получилось

- ☑ Параметр скрыт в SparkUI
- ☑ Параметр скрыт в логах драйвера
- ☑ Параметр скрыт в ps на хосте запуска spark-submit

Но в списке процессов в PySpark…

- Пользователь PySpark в Jupyter заметил в своём списке процессов какую-то джаву с токеном в открытом виде

```
> ps aux | grep spark

python …

java -cp … org.apache.spark.deploy.SparkSubmit
  --conf spark.yt.token=token …
```


HL HighLoad++ Весна 2021

Прокидывание секретов

- ☑ Параметр скрыт в SparkUI
- ☑ Параметр скрыт в логах драйвера
- ☑ Параметр скрыт в ps на хосте запуска spark-submit
- ✗ Параметр виден в ps при запуске PySpark

HL HighLoad++ Весна 2021

Что произошло?

- В коде для питона я прокидывала параметр при старте сессии

```
// example.py

spark = SparkSession.builder
                    .config("spark.yt.token", token)
                    .getOrCreate()
```


HL HighLoad++ Весна 2021

Что произошло?

- В коде для питона я прокидывала параметр при старте сессии
- Но в PySpark вызов getOrCreate запускает jvm с SparkSubmit

```
// example.py
spark = SparkSession.builder
                    .config("spark.yt.token", token)
                    .getOrCreate()
```


HL HighLoad++ Весна 2021

Что делать?

- У нас уже есть костыль в SparkSubmit, можем им воспользоваться

```
// example.py

os.environ["SPARK_YT_TOKEN"] = token
spark = SparkSession.builder.getOrCreate()
```


HL HighLoad++ Весна 2021

Что получилось

- ☑ Параметр скрыт в SparkUI
- ☑ Параметр скрыт в логах драйвера
- ☑ Параметр скрыт в ps на хосте запуска spark-submit
- ☑ Параметр скрыт в ps при запуске PySpark

Но в кластере на воркере…

- В списке процессов воркера, на котором запущен драйвер, видны все аргументы драйвера

```
> ps aux | grep spark

java -cp ... -Dspark.yt.token=token …
org.apache.spark.deploy.worker.DriverWrapper …
```


HL HighLoad++ Весна 2021

Что получилось

- ☑ Параметр скрыт в SparkUI
- ☑ Параметр скрыт в логах драйвера
- ☑ Параметр скрыт в ps на хосте запуска spark-submit
- ☑ Параметр скрыт в ps при запуске PySpark
- ✕ Параметр виден в ps на воркере в кластере

HL HighLoad++ Весна 2021

Что делать?

- Я добавила костыли
  - В запуск драйвера в SparkStandalone
  - В инициализацию SparkConf

Что получилось

- ☑ Параметр скрыт в SparkUI
- ☑ Параметр скрыт в логах драйвера
- ☑ Параметр скрыт в ps на хосте запуска spark-submit
- ☑ Параметр скрыт в ps при запуске PySpark
- ☑ Параметр скрыт в ps на воркере в кластере

Что делать?

- Не прокидывать секреты через конфиг спарка. Никогда. Даже если кажется, что это просто
- Использовать специальные сервисы для хранения секретов

Итоги

- Простое решение было сделано за 3 дня
- Рефакторинг на более сложное занял ещё несколько недель
- Чтение батчами ускорило запросы в 2-3 раза

В следующей серии…

- Pushdown predicate
- Оптимизация записи
- Использование информации о сортированности данных

- Гибкая интеграция с шедулером

HL HighLoad++ Весна 2021

Спасибо! Вопросы?